This dissertation/thesis/report entitled:

written by:

has been approved for the degree of:

in

by

1 - Advisor Name

2 - Department Chair or Committee Member Name

3 - Committee Member Name

4- Committee Member Name

5 - Committee Member Name

6 - Committee Member Name

Month YEAR

Bespoke Neural Network Architectures for Rapid Multivariate Time Series Classification and Representation

by

Alexander Michael Moore

A Dissertation

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree of Doctor of Philosophy

in

Data Science

February 22, 2023

Professor Randy Paffenroth
Worcester Polytechnic Institute
Advisor

Professor Mohamed Eltabakh
Worcester Polytechnic Institute
Committee Member

Professor Oren Mangoubi
Worcester Polytechnic Institute
Committee Member

Dr. Joshua Uzarski
U.S. Army DEVCOM Soldier Center
External Committee Member

Abstract

Accurate chemical sensors are vital in medical, military, and home safety applications. Training machine learning models to be accurate on real world chemical sensor data requires performing many diverse, costly experiments in controlled laboratory settings to create a data set. In practice even expensive, large data sets may be insufficient for generalization of a trained model to a real-world testing distribution. This dissertation is concerned with the application of modern machine learning and deep learning techniques to a real-world, low-data chemical sensing task. In order to mitigate the challenges of an application with costly data, we develop algorithms in adversarial learning and data synthesis, regularize models with multitask and multi-loss learning, and transfer knowledge between multiple domains such that the ultimate goal of chemical detection is improved. We include novel research on data sets within the chemical sensing as well as natural image and molecular representation literature. Machine learning and deep learning models have been adapted with novel architectures from tabular, time series, and natural image domains which ultimately improve downstream classifier performance.

Acknowledgements

I am overwhelmingly grateful to the large team of my advisors and peers which have brought me to this stage of my PhD. I foremost thank my advisor Professor Randy Paffenroth.

I would not be presenting research if not for our collaborators and foremost surface chemistry experts Dr. Joshua Uzarski and Dr. Ken Ngo, who at every turn took on additional challenges to develop our research. Without their expertise, this work would not exist.

I would like to thank my WPI committee members, Professor Eltabakh and Professor Mangoubi, for donating their time, energy, and advisement through research collaboration and the defense.

Finally I would like to show my gratitude to my many WPI peers and collaborators. Collaboration and critical conversation with my incredible peers has made me the excited and curious machine learning researcher I am today.

Papers Contributing to this Proposal

Published First Author Papers

A. Moore, R. Paffenroth, K. Ngo, J. Uzarski, ”Cycles Improve Conditional Generators: Synthesis and Augmentation for Data Mining,” Advanced Data Mining and Applications (ADMA), 2022.

A. Moore, R. Paffenroth, K. Ngo, J. Uzarski, ”ACGANs Improve Chemical Sensors for Challenging Distributions,” IEEE International Conference on Machine Learning and Applications (ICMLA), 2022.

Published Second Author Papers

E. Prihar, **A. Moore**, N. Heffernan, ”Identifying Struggling Students by Comparing Online Tutor Clickstreams,” Artificial Intelligence in Education, 2021.

E. Prihar, **A. Moore**, N. Heffernan, ”Identifying Explanations Within Student-Tutor Chat Logs,” International Conference on Educational Data Mining, 2022.

Under Review

In addition to papers currently in print at conferences and machine learning journals, this dissertation will also cover the following material pending conference or journal submission. Each of these papers is complete and awaiting conference or journal acceptance or pending Department of Defense approval for release:

A. Moore, R. Paffenroth, K. Ngo, J. Uzarski, ”ChemVise: Maximizing Out-of-

Distribution Chemical Detection with the Novel Application of Zero-Shot Learning," Available on the arXiv[1] pending admission to ICMLA 2023.

A. Moore, R. Paffenroth, K. Ngo, J. Uzarski, "ChemTime: Confident Early Multivariate Time Series Classification for a Novel Chemiresistive Sensor Array," pending Department of Defense approval for release.

[1] http://arxiv.org/abs/2302.04917

Executive Summary

Dissertation Keystones

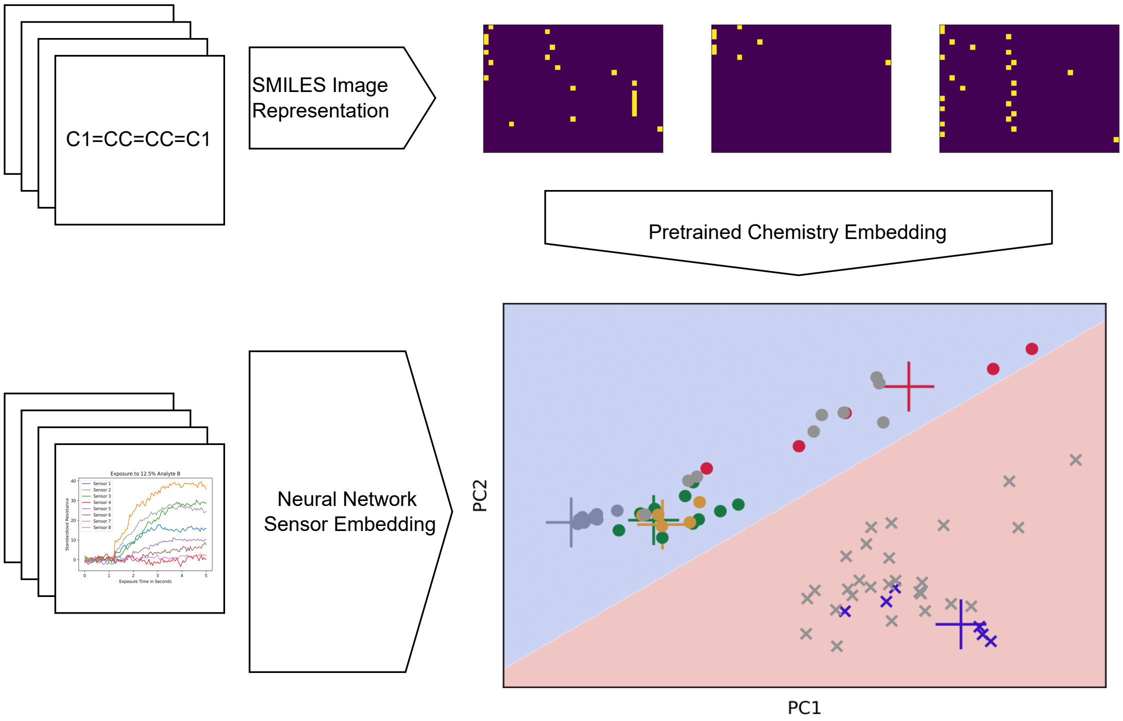

C1=CC=CC=C1
SMILES Image Representation
Pretrained Chemistry Embedding
Exposure to 12.5% Analyte B
Neural Network Sensor Embedding
PC2
PC1

Figure 1: A preview of the *ChemVise* approach (Chapter 4). A modification to the transfer learning protocol allows us to incorporate chemistry knowledge for a novel hardware sensor array. Extensions to *ChemVise* lead to *ChemTime* (Chapter 6), which facilitates classification and inference using representation sequences in an information-dense space.

This dissertation develops novel algorithms for the representation of natural images and sensor data. Learning effective representations for downstream supervised, unsupervised, and synthesis tasks is essential to modern deep learning application. Transfer learning (Weiss et al., 2016) is a powerful approach to leverage data from related domains between tasks by sharing parameters of deep learning models. For a real-world novel chemiresistive array, however, there does not exist a set of neural network parameters which can handle encoding data from an unknown hardware sensor array. We therefore turn to the literature on semantic embeddings of class targets from the natural image literature with a novel extension to molecular representation (Ramesh et al. (2022), Chapter 4).

For chemical sensing, compute-efficient parameterized representations of chemical sensor resistances in a chemistry-informed space improves inference, reduces detection time, and builds explainability into classification outcomes. Included in this dissertation are multiple approaches to multitask and adversarial learning algorithms for bespoke neural network architectures on downstream applications including natural image synthesis and classification, and early multivariate time series classification (Figure 2, Chapter 6).

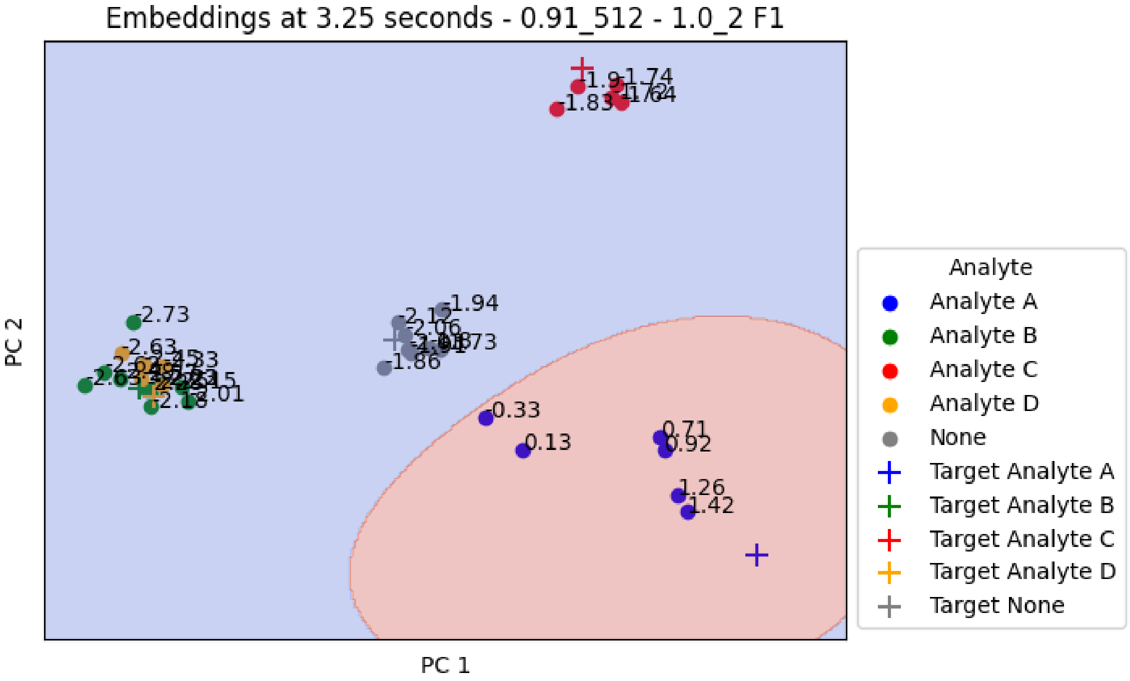


Figure 2: Chapter 6 will extend the *ChemVise* algorithm to *ChemTime* by incorporating inference and early classification for sequences of learned representations.

Our Contributions

- We induce reconstructive cycles and multitask learning for a natural image synthesis task. We demonstrate not only can baseline conditional GANs be improved by additional losses without requiring additional data or labelling, but data synthesis further improves supervised learning outcomes on auxiliary classification models. Data synthesizing models and their advantages to downstream representations are explored in Chapter 2.
- We are the first to apply multitask learning with adversarial data synthesis losses to a real-world chemiresistive sensor array data set. We demonstrate that multitask-loss supervised neural networks from *Cycles Improve Conditional Generators* substantially improved classification outcomes for an obscured set of chemical analytes shown in Chapter 3. Extensions to this finding lead to further inquiry on combining generative cycles with downstream time series classifiers in Chapter 6.
- We develop a novel architecture and extension of visual-semantic embedding research (Frome et al., 2013; Ramesh et al., 2022) in the natural image domain to a chemistry application (Chapter 4). The novelty of *ChemVise* comes from modifying transfer learning to map chemiresistive sensor array data from a bespoke sensing device to a generic molecular representation space. Translating chemiresitive data to pretrained representation spaces shapes the ongoing direction of this research. Extensions to the *ChemVise* paradigm to multivariate time series classification are explored in Chapter 6.
- We design the first empirical study of the machine learning literature on multivariate hardware sensor arrays for the classification of chemical analytes

based on the blueprint from Ruiz et al. (2021). We benchmark over 30 multivariate time series classification algorithms with a novel hardware sensor array to determine appropriate algorithms for a real-world deployed tool. Our research in Chapter 5 expands the set of classifiers beyond the scope of other similar studies on multivariate time series, and alters the scope of research to focus on optimization of sensor array chemistries with unique binding affinities to maximize analyte discriminability.

- We propose, implement, and study the first semantic-representation model for the early classification of chemical analytes. Chapter 4 demonstrates how *ChemVise* meaningfully embeds target analytes for arbitrary chemical sensing hardware data. *ChemTime* (Chapter 6) will leverage these chemistry-informed latent spaces with the signal-translation model utilizing contemporary machine learning to provide explainable, inferential classifications of representation sequences.

- We introduce and benchmark the novel architecture and training algorithm *ChemTime*, with multiple experimental results demonstrating substantial improvement in rapid classification, early classification, and signal efficiency for a real-world detection task against a peer group of highly competitive and recently published multivariate time series classifiers.

Publication Web

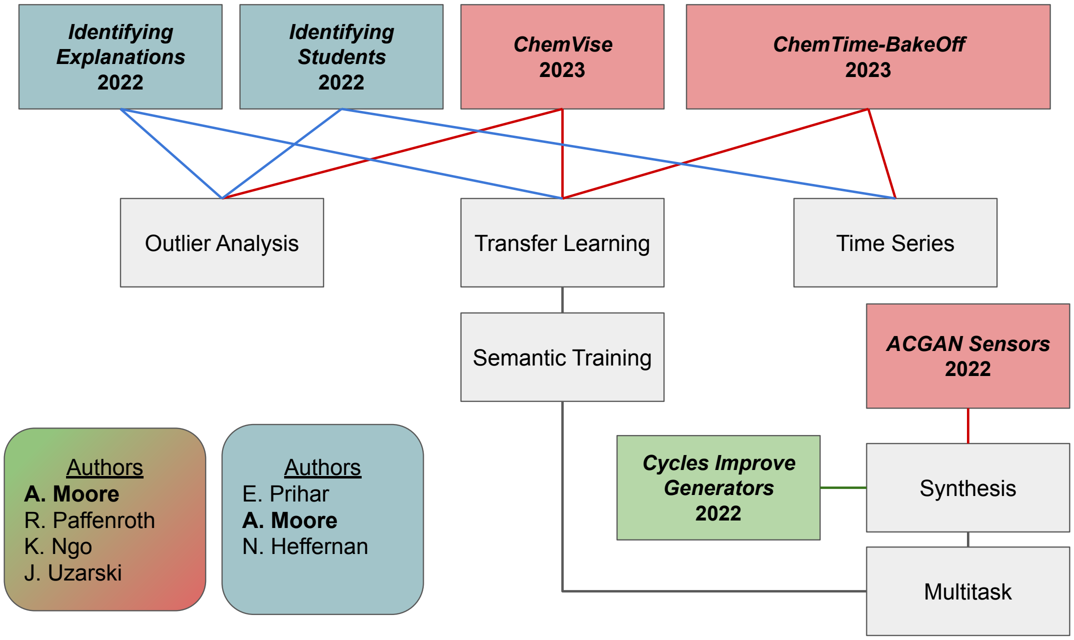


Figure 3: Here we summarize the interconnectedness and diversity of our published and publish-pending publications. In color, we denote our primary domains of application: AI in education, natural images, and chemical sensing. In blue, *Identifying Struggling Students* and *Identifying Explanations in Chatlogs* apply outlier analysis and transfer learning of large language models to promote student learning in an online platform. In green, our development of multitask reconstructive conditional GANs for natural images in *Cycles Improve Conditional Generators*. *ChemTime-BakeOff* in red combines our first application of multivariate time series classifiers to real-world hardware sensing data (formerly *Chemical Sensing Bake Off*) with *ChemTime*, the extension of the *ChemVise* approach to multivariate time series for rapid and early classification using representation sequences. *ChemTime-BakeOff*, *ACGANs Improve Chemical Sensors*, and *ChemVise* in red are our publications on the application of novel deep learning architectures with semantic and multitask training to chemiresistive sensor data.

Contents

Notation

This dissertation makes use of mathematical notation for a variety of data descriptions and model training algorithms. Here, we standardize notation for at-a-glance recognition:

Object	Notation
Real value	x
Vector	$\vec{x}$
Matrix	X
Array	$\mathbb{X}$
Machine Learning Model	$\mathbf{X}$

1 Preliminaries

Extensive literature exists for the applications of time series classification (Ruiz et al., 2021), deep representation learning (Weiss et al., 2016), and chemical sensor classification (Zhang et al., 2021) - though bridging between these domains to improve real-world chemical detectors remains unexplored. In this chapter we will motivate each element of the synthesis and introduce topics which will be subsequently explored. Section 1.1 introduces the topic of representation learning to improve downstream task performance. Section 1.3 and 1.4 will introduce the unique domain of chemical sensors and the method with which the data used throughout multiple chapters are gathered. Section 1.5 will introduce the application of machine learning to time series classification.

1.1 Representations of Data

The "depth" of deep learning refers to the iterative layer functions applied to input data in a neural network model. Successive neural network layers compose a linear projection with a simple nonlinear activation function such as pooling or element-wise hinge functions. An emergent behavior within these layers which has revolutionized artificial intelligence and machine prediction is representation learning (Bengio et al., 2013). Successive neural network layers are capable of learning abstractions of input data which facilitates further transformations of subsequent layers. In natural image processing, convolutional filters may learn to detect lines, then edges, then abstract shapes given sufficient model capacity (Krizhevsky et al., 2012). Deep neural networks trained to predict the class of an input sample implicitly learn representations due to *representation bias* (Caruana, 1997). Representation bias implies that neural network losses are minimized when

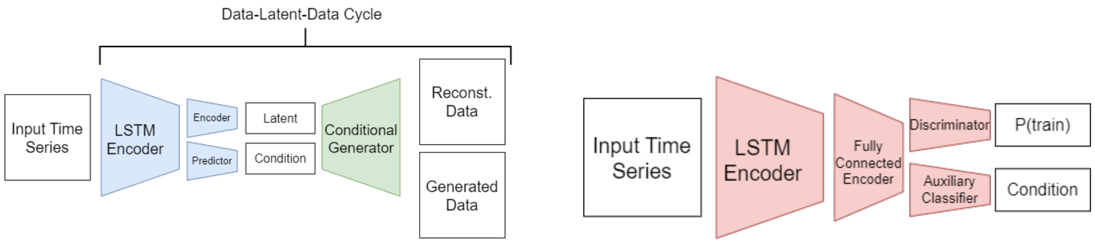


Figure 4: The AC-T-CAEGAN will be built iteratively in the following chapters on multitask adversarial training for data synthesis, chemistry representation spaces, and time series representations. The CAEGAN (Chapter 2) and ACGAN (Chapter 3) are combined into one training procedure with modified time series architecture from Chapter 6.

models learn representations of input samples generalizable to multiple classes.

Perhaps the most revolutionary idea of contemporary deep learning is an exploitation of representation bias called transfer learning (Weiss et al., 2016). Transfer learning utilizes parameterized representations from a deep neural network trained under a different loss function or data set to encode samples to meaningful vector embeddings. Explosive results in modern artificial intelligence and deep learning rely on the transfer of representations, including ChatGPT and GPT-3 (Brown et al., 2020), DALL-E 1 and 2 (Ramesh et al., 2021, 2022), and Stable Diffusion (Rombach et al., 2021).

Each chapter of this dissertation proposal builds on the concept of learning representations in different ways. Multitask learning induced through reconstructive and adversarial losses demonstrates that the representation bias improves the generalizability of parameterizations of neural networks (Chapters 2 and 3). Subsequent chapters will extend transfer learning of representations to transfer learning for semantic target embeddings (Chapter 4, Frome et al. (2013)). Figure 4 shows a hypothetical combination of multiple approaches to improving representations, from adversarial losses on sequences of representations learned by a time series embedding model.

1.2 Representations of Molecules

Partially due to the meteoric rise of deep learning approaches for natural image and natural language processing, many practitioners are intensely researching the application of machine and deep learning to applied chemistry which was originally met with skepticism by chemists practicing hand-coded heuristics such as the Morgan Fingerprint (Mater and Coote, 2019; Morgan, 1965). Researchers have seen rising application of machine learning to the physical sciences, with "many demonstrating significant improvements in predictive accuracy and ability to replicate human decision making" (Mater and Coote, 2019; Chen et al., 2018). For an excellent review of deep learning applications to chemistry and the reception history of chemical applications, please refer to (Mater and Coote, 2019) as well as (Schmidt et al., 2019) for an excellent survey of surface chemistry. For an introduction to machine learning and its applications for chemists we recommend (Mueller et al., 2016).

A primary concern of computation for applied chemistry is the representation of chemical analytes in machine-readable format. This topic has been relevant to chemical researchers for over 160 years - long before even the advent of electronic computation (David et al., 2020; Wiswesser, 1968). In the deep learning field as in chemistry, representations must serve the purpose of encoding meaningful elements of the object they represent and for machine and deep learning must be processable with vector mathematics. The question of representation poses as many challenges to the encoding of molecules and compounds as it does to images and written text. No singular representation will be appropriate for every task, so countless molecular descriptors, embeddings, and encoding techniques have been proposed to compete on a variety of downstream supervised tasks using transfer learning (Weiss et al., 2016).

One such chemical representation is SMILES (Weininger, 1988), which encodes molecular structure to ASCII strings given by depth-first traversal over molecular bonds. While the SMILES approach does represent molecules in a machine-readable manner, deep learning researchers have recently relied on the implicit *representation learning* that occurs in multi-layer networks as the source of information density in vector representations (Bengio et al., 2012). The feedforward deep neural network outlined in Chemception (Goh et al., 2017) was trained on a set of molecular images derived from SMILES strings (Figure 5) to predict the toxicity, activity, and solvation properties of the input. As a consequence of implicit representation learning in deep neural networks, the penultimate activations of this model may be used to represent molecular analytes in dense real-valued vector embeddings. The Chemception embeddings encode semantically meaningful distances between points which facilitate vector mathematics and downstream learners for fine-tuned tasks (Chapter 4). Chemception and Mol2vec (Jaeger et al., 2017) use modern deep learning practice to outperform the earlier Morgan Fingerprints representation using pattern recognition on a massive corpus of chemistry data.

Contemporary research utilizes deep learning for chemistry, and in particular compound representations for supervised learning such as drug interaction (Ryu et al., 2018; Cai et al., 2019), new material discovery (Wei et al., 2019; Schmidt et al., 2019), and drug design (Yang et al., 2019; Gebauer et al., 2022; Elton et al., 2019; Rifaioglu et al., 2020). These approaches utilize pretrained or novel deep learning embedding models to learn encodings given a large set of example molecules. The encodings are used as information-dense representations of training data for an

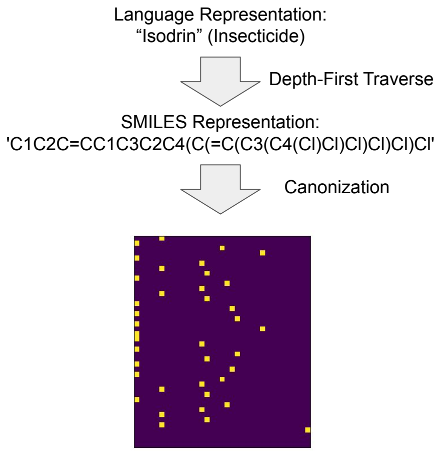
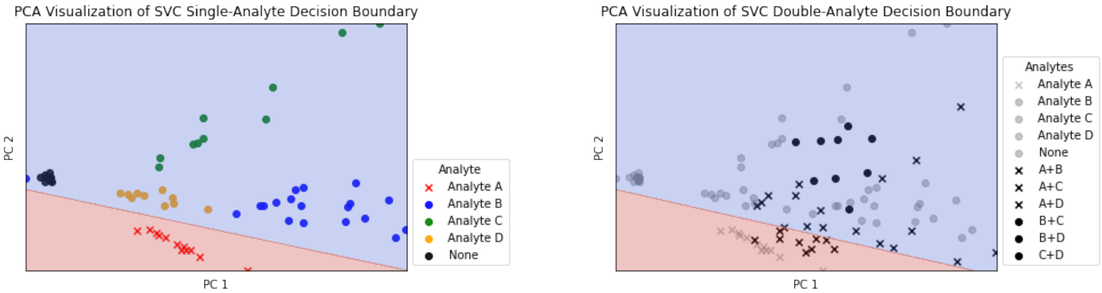


Figure 5: A simple but flawed approach to preparing molecular data for convolutional neural network training is converting molecular SMILES to images.

external task outside of the initial scope of the embedding model.

Not all representations are created equal. Figure 6 shows how simple approaches to the representation of chemiresistive signal data may fail by overlaying challenging samples on a linear decomposition approach. Classifiers trained to classify a chemical analyte in this example experiment fail to generalize to obscured hold-out samples due to an overly simple encoding model. Learning representations of complex data is a recurring theme of Chapters 2, 3, 4, and 6. Though deep learning models learn parameterized representations of data which may be used as inference as in Goh et al. (2017) and Soloveitchik et al. (2021), explicit domain knowledge may be modelled in representation spaces which beget greater inferential capabilities. This dissertation proposal will emphasize transfer learning for structured representation spaces, and discuss how these meaningful embeddings can incorporate research from the machine learning and chemical sensing literature. We will emphasize confidence, inference, and out-of-distribution capabilities through representation learners on chemiresistive signals which traditional machine learning classifiers or black-box deep learning approaches may lack.



(a) (b)

Figure 6: Visualization of the shortcomings of linear decomposition with linear classifiers for mixture data. **Left**: Under a 2-dimensional PCA transformation, signal data of single-analyte exposures may be well-separated by a linear classifier. **Right**: Under a PCA embedding of signal data, the same model fails to accurately discriminate between samples containing Analyte A. Samples marked with x contain Analyte A and are intermixed with non-Analyte A single-analyte exposures, leading to poor classification results from this linear classifier.

1.3 Chemical Sensing

Chemical sensors measure physical and chemical properties of analytes into measurable signals. Examples of chemical sensors include breathalyzers, carbon monoxide sensors, and electrochemical gas sensors (Rana et al., 2021). The detection of particular chemical analytes is highly relevant in civilian safety, manufacturing, and military applications (Weiss et al., 2018; Wiederoder et al., 2017). Chemiresistive sensors respond to chemical analytes by reporting changes in resistance through a coated resistor. Chemical analytes interact at the molecular level by bonding with the sensor coating, called adsorption. Analyte adsorption to the sensor coating causes the resistance through the sensing element to change as a function of the binding affinity between the analyte and surface. The binding affinity between analytes and coatings is affected by their molecular and polymer chemical properties.

Many research groups have invested in developing parallel sensor arrays with diverse approaches to gas discrimination (Park et al., 1999; Tomchenko et al., 2003; Nallon et al., 2016; Zhang et al., 2021). There has been increasing interest in fast, accurate, and ”highly sensitive gas sensors with excellent selectivity” in part due to high demand for environmental safety and healthcare applications and significant research has been conducted to investigate the application of highly selective materials (Wiederoder et al., 2019; Yaqoob and Younis, 2021). Wiederoder et al. (2017) address the challenges of rapid discrimination of multiple target analytes with low-cost, low-power, miniaturized sensors capable of analyte detection despite obscurants with ”an array of semi-selective chemical sensors that respond to many analytes simultaneously” which creates ”unique analyte signatures for detection and classification of multiple analytes” on a single platform.

1.4 Chemical Sensing Data

Research included in this dissertation proposal builds machine and deep learning classifiers on data drawn from an 8-sensor chemiresistive array with chemically diverse coatings to maximize analyte discriminability as in (Nallon et al., 2016; Wiederoder et al., 2017; Weiss et al., 2018). The data and results of subsequent chapters call upon several experiments performed with various hardware chemiresistive sensor arrays, but the deep learning methodology for transferring analyte representations and supervised classification of multiple analytes may be applied to any chemical sensor array. Concepts surrounding transfer learning and semantic representations (Chapters 4, 6) can be extended to any data domain with appropriate choice of target embeddings.

Chemiresistive chemical sensors utilize resistance changes of electrodes with unique coatings. Different vapor analytes have different adsorption affinities to the different coatings, resulting in discriminative resistance changes adaptable for various chemical analytes. Figure 7 visualizes the temporal regions of a full analyte exposure experiment. Analyte exposure experiments utilizing chemiresistive sensors contain three exposure periods (Nix, 2022):

1. Controlled flux of analyte and adsorption to the active surface.
2. Change in electrical resistance due to interaction of sensor coating with adsorbed or absorbed analyte molecules.
3. Cessation of analyte flux followed by flux of zero gas (Nitrogen). Desorption (departing) of analyte from sensors which reverses resistance change.

To collect one experimental trial of the data sets used in subsequent chapters, the sensors first rest for a baseline period of ten seconds. This period is used to calculate the mean shift in sensor resistance to standardize the data during preprocessing. A controlled flow of analyte vapor is released into the sensor chamber for a thirty second adsorption response time before the exposure valve is shut. Finally, a desorption time begins and lasts forty seconds in which the sensor resistances recover and no further analyte vapor is released, indicated in Figure 7 by the lengthy period of gradual decline in resistance. Between trials, an eight minute recovery period allows the sensors to fully recover and any lingering analyte vapor to dissipate. Triplicate trials for each analyte concentration are performed for data integrity.

The unique coatings on the chemiresistive sensor cause lead to characteristic resistances of analytes which facilitate discrimination of the gas exposures. Figure 8 shows an example of the contrast in sensor responses to 17.5% Analyte A and

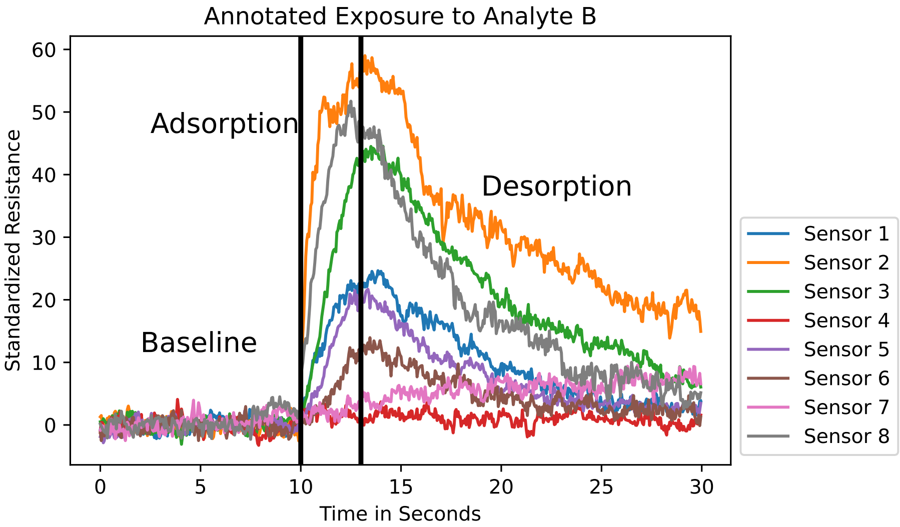


Figure 7: One experimental sample including baseline, absorption, and desorption for an exposure of Analyte B. The sensor resistances rapidly increase during the analyte exposure period as chemical analytes adsorb onto sensor coatings before returning to baseline levels during desorption as analytes diffuse from the surface.

17.5% Analyte B vapors given the same set of eight sensors with bespoke chemiresistive coatings. From the 8-channel characteristic signal machine learning models learn decision patterns for generalization to unseen testing samples.

1.5 Time Series Classification

Time series data are sequences of features with some inherent ordering, such as stock closing prices for each day of a year. The structural ordering of the features affects the discriminability of samples in the same manner as neighboring pixels in a natural image determine meaning to a human viewer (Bagnall et al., 2017). This ordering does not need to be through time but could be through frequencies, magnitudes, or observations along any ordered axis.

Time series classification is highly relevant in applications across all domains in the physical sciences, manufacturing industries, technology, and more. From sen-

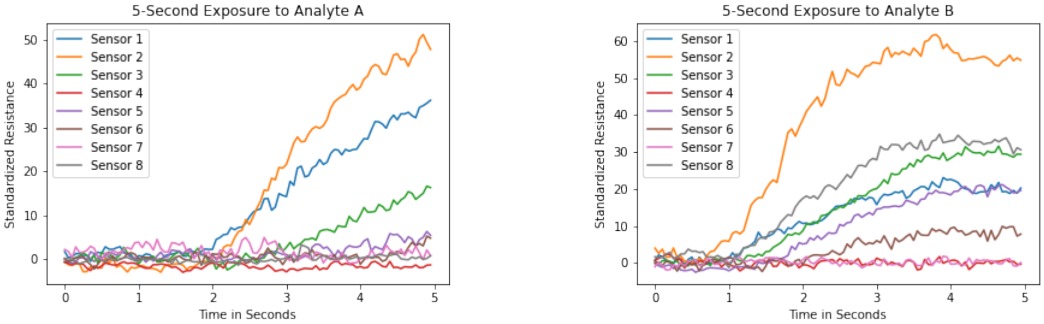


Figure 8: Two five second single-analyte exposures to different analytes at the same concentration. Discrepancies in sensor resistance are explained by adsorption interactions between analytes and sensor coatings.

sors (Weiss et al., 2018), video analysis Jönsson and Eklundh (2004), computer network traffic analysis (Madan and Mangipudi, 2018), biomedical monitoring (Lin et al., 2005), manufacturing (Li et al., 2011), and airline industry efficiency (Semenick Alam and Sickles, 2000), time series may be the most prolific and diverse form of data humans regularly create, interact with, and learn from (Gupta et al., 2020). The inherent information in the ordering of features makes time series a foundational starting point of data analysis applications which must capture dependencies through time, rather than considering an observation as a collection of features.

One univariate time series observation s is a sequence of ordered pairs $(timestamp, value)$ (Xing et al., 2012). This proposal extends machine learning for time series to multivariate time series (MTS). MTS data contain multiple features at each timestep of a signal, where the time series is a list of vectors over d dimensions and n observations. Data set $\mathbb{X}$ is given by observations $< X_1, ..., X_n >$. The t-th time index of the i-th sample of dimension k is the scalar $x_{i,t,k}$ (Ruiz et al., 2021). *Early* and *rapid* classification of multivariate time series (Chapters 3, 5, and 6) will call upon subsequences of $\vec{x}$. The *length-l* prefix subsequences of sample X_i are $X_i[:, 1, t]$, or the first t time observations of sample X_i for all features. Subsequences are sometimes called *shapelets* in the literature and are used as discriminative sub-elements of signals with distance metrics (Bostrom and Bagnall, 2017).

2 Data Synthesis

2.1 Generative Adversarial Networks

Generative Adversarial Networks (GANs, (Goodfellow et al., 2014)) learn a generative function with parameters mapping from a low-dimensional standard normal prior to an estimate of the training distribution in the data space based on a game-theoretic min-max game. Two deep learning agents, the generator **G** and discriminator **D**, alternate weight updates to maximize the adversary agent error.

GANs have received extensive research for their promising capabilities to learn approximations to training data distributions given a min-max game. The min-max optimization uses expectation over samples from the training set $X \sim p_{data}$ of the error of the discriminator given by $\log \mathbf{D}(X)$. The discriminator error on real samples is balanced against that of error on generated samples given by the expectation over samples $\vec{z} \sim p_z(z)$ decoded by the generator $\mathbf{G}(\vec{z})$, where z is the multivariate standard normal prior. The agents update in turn, where the generator tries to minimize the objective against a maximizing discriminator solved by

$$\begin{aligned} L_{GAN} = \min_{\mathbf{G}} \max_{\mathbf{D}} V(\mathbf{D}, \mathbf{G}) = \mathbb{E}_{\mathbb{X}}[\log \mathbf{D}(\vec{x})] \\ + \mathbb{E}_{\vec{z}}[\log 1 - \mathbf{D}(\mathbf{G}(\vec{z}))]. \end{aligned} \tag{1}$$

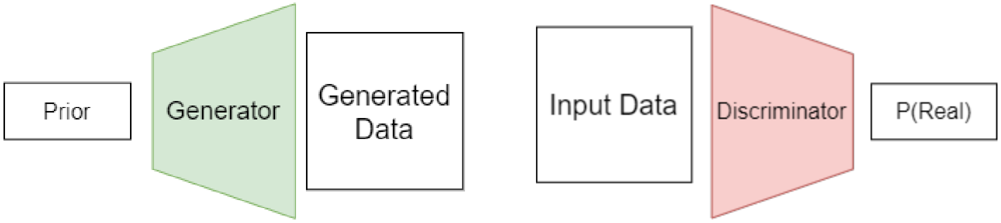
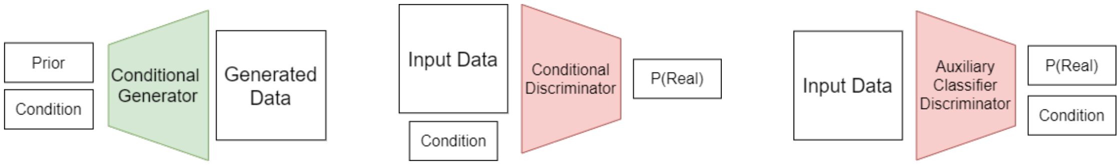


Figure 9: GAN Architecture.



(a) The conditional generator.

(b) The conditional discriminator.

(c) The auxiliary classifier discriminator.

Figure 10: Baseline conditional GAN architectures.

2.2 Conditional GANs

Conditional GANs (Figure 10) impart structure on the inputs to the generator and discriminator by concatenating some label $\vec{y}$ to the input vectors of each model (Mirza and Osindero, 2014). This conditions the min-max optimization as:

$$\begin{aligned}\min_{\mathbf{G}} \max_{\mathbf{D}} V(\mathbf{D}, \mathbf{G}) = {} & \mathbb{E}_{\mathbb{X}}[\log \mathbf{D}(\vec{x}|\vec{y})] \\ & + \mathbb{E}_{\vec{z}}[\log 1 - \mathbf{D}(\mathbf{G}(\vec{z}|\vec{y}))].\end{aligned} \tag{2}$$

2.3 Auxiliary Classifier GANs

The Auxiliary Classifier GAN (ACGAN) modifies the concatenated-input discriminator of the CGAN to instead predict the corresponding class of input training and synthetic samples (Odena et al., 2017). This modification is discussed in Section 2.4 as a potential benefit of training mixed-loss adversarial models. The ACGAN alters GAN training by incorporating an auxiliary classifier head on the output of the discriminator, with a conditional component given by:

$$L_C = \mathbb{E}_{\mathbb{X}}[\log P(C = \vec{c})] + \mathbb{E}_{\vec{z}}[\log P(C = \vec{c})]. \tag{3}$$

Using this conditional loss, the discriminator is trained to maximize $L_{GAN} + L_C$ (1) while the Generator is trained to maximize $L_C - L_{GAN}$ (Odena et al., 2017). In addition to producing labeled data by specifying conditions, approximating conditional distributions of training data with conditional generators is demonstrated to improve generative quality and discriminability of samples over unconditional GANs when labels are available (Odena et al., 2017).

2.4 Cycles Improve Conditional Generators

Beyond demonstrations producing high-quality synthetic images, training generative models may improve downstream data efficiency, generalization, and robustness of deep learning models across many domains (Child, 2021). Generative Adversarial Networks (GANs) have received extensive ongoing study since their inception, and extensions proposed in the seminal introduction of GANs hinted at the subsequent development of conditional GANs which specify the desired class of generated samples (Goodfellow et al., 2014; Mirza and Osindero, 2014). Conditional image synthesis with GANs learns functional approximations from a joint prior-condition space to a joint condition-data natural image space given labeled training samples (Mirza and Osindero, 2014; Odena et al., 2017).

Accounting for conditional distributions in the training of data synthesis models allows for selection of generated data classes which may improve data efficiency and model robustness during downstream training (Section 2.14, (Child, 2021)). The utilization of downstream supervised learners may additionally be a promising metric in the training of conditional generators, particularly in non-natural image tasks which lack standards of evaluating synthesized samples (3).

This chapter proposes alterations to the CGAN architecture and training procedures which improve upon baseline conditional image synthesis according to a variety of established and proposed evaluation metrics with a corresponding incremental study isolating contributions to model improvement. We demonstrate that incremental improvements on conditional image synthesis as measured by generative quality metrics correlate to improved downstream learner accuracy for models trained on synthesized samples (2.14). This finding begets ongoing study of the role of conditional data synthesizers for data set augmentation to improve model robustness, data efficiency, and performance on low-data paradigms.

We study three models which improve upon baseline conditional generative GANs by inducing cycles inspired by unpaired image-to-image translation from CycleGAN (Zhu et al., 2017) with extensions to conditional data generation (Section 2.4). We define cycles as a composition of functions from an initial space to an intermediate space and back with low reconstructive error such that each function image fulfills some distributional requirement: in the case of CycleGAN ((Zhu et al., 2017), 2.6), both spaces are image spaces with an adversarial discriminator. By setting one of these spaces to be a multivariate normal latent prior, we induce an autoregressive model with an added adversary, here called the conditional autoencoder-GAN outlined in Section 2.8.

Two incremental modifications to this design are introduced in Sections 2.9 and 2.10 which isolate the contributions of cycles versus autoencoding for encoder-decoder models. A study comparing equivalent baseline conditional GANs to cyclical models is performed in Section 2.12. We find that enforcing conditional, reconstructive, and cyclical losses on the proposed models improves image synthesis outcomes.

2.5 Contributions

This chapter contributes to the improvement of conditional natural image synthesis with GANs and the utilization of conditional generation for simultaneous or downstream supervised learning as follows:

1. A novel formulation of three conditional cyclical GANs to incorporate cycling between spaces as well as a bipartite latent space for conditioning (Section 2.4).

2. An incremental study on the inclusion of cycles to conditional generators demonstrating improvement on baseline conditional generators across a variety of experiments and metrics (Section 2.11).

3. A proposed utilization of conditional image synthesis for supervised learning data set augmentation as an alternative generation metric (Section 2.14).

2.6 CycleGAN

The family of models proposed in this chapter draws from the CycleGAN unpaired image-to-image translation model (Zhu et al., 2017). CycleGAN (Figure 11) training uses two unpaired image sets drawn from distinct training distributions X and Y, typically natural images. Two models comparable to GANs are trained simultaneously: One generator learns $\mathbf{F}(\vec{x}) = \hat{\vec{y}}$ for training data $\vec{x} \in X$ directed by a discriminator trained to distinguish training from synthesized samples in Y-space, and another function which learns $\mathbf{G}(y) = x$ for training data $y \in Y$ taught by a discriminator who learns to distinguish training from synthesized samples in Y-space. These generators are trained under the *cycle* constraint enforced by the reconstructive penalty $\mathbf{F}(\mathbf{G}(\vec{y})) \approx \vec{y}$ and $\mathbf{G}(\mathbf{F}(\vec{x})) \approx \vec{x}$ (Equation 6).

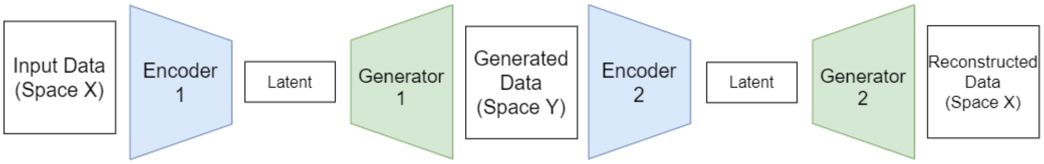


Figure 11: CycleGAN Architecture. Each space X, Y is accompanied by a discriminator returning the adversarial generation loss Equation 1.

$$\begin{aligned}\mathcal{L}_{GAN}(\mathbf{G}, \mathbf{D}_Y, X, Y) = \mathbb{E}_{y\sim p_{data}(y)}[\log \mathbf{D}_Y(y)] \\ + \mathbb{E}_{\mathbb{X}}[\log \mathbf{D}_Y(\vec{x})]\end{aligned} \tag{4}$$

Given generative models $\mathbf{G}, \mathbf{F}$ with images in spaces X, Y respectively and discriminators $\mathbf{D}_X, \mathbf{D}_Y$ on spaces X, Y respectively, the full optimization objective $\mathcal{L}(\mathbf{G}, \mathbf{F}, \mathbf{D}_X, \mathbf{D}_Y)$ is given by:

$$\mathcal{L}_{GAN}(\mathbf{G}, \mathbf{D}_Y, X, Y) + \mathcal{L}_{GAN}(\mathbf{F}, \mathbf{D}_X, \mathbf{Y}, \mathbf{X}) + \lambda \mathcal{L}_{cyc}(\mathbf{G}, \mathbf{F}) \tag{5}$$

Where

$$\begin{aligned}\mathcal{L}_{cyc}(\mathbf{G}, \mathbf{F}) &= \mathcal{L}_{recon}(\mathbf{F}, \mathbf{G}, X) + \mathcal{L}_{recon}(\mathbf{G}, \mathbf{F}, Y) \\ &= \mathbb{E}_{\mathbb{X}}[||\mathbf{F}(\mathbf{G}(\vec{x})) - \vec{x}||_1] \\ &+ \mathbb{E}_{Y}[||\mathbf{G}(\mathbf{F}(\vec{y})) - \vec{y}||_1].\end{aligned} \tag{6}$$

The hyperparameter λ tunes the relative importance of reconstruction and adversarial components. This system optimizes the following:

$$\mathbf{G}^*, \mathbf{F}^* = \arg\min_{\mathbf{G},\mathbf{F}} \max_{\mathbf{D}_x,\mathbf{D}_y} \mathcal{L}(\mathbf{G}, \mathbf{F}, \mathbf{D}_X, \mathbf{D}_Y). \tag{7}$$

Though CycleGAN emphasizes natural image style transfer, this research investigates the usage of cycles between an unpaired multivariate normal latent space and a natural image data space for conditional synthesis and demonstrates that the regularization imparted by cycles may improve training outcomes over non-cyclical baseline models via regularization imparted by additional optimization objectives.

2.7 Cycles for Conditional Generation

This research proposes the Conditional Autoencoder-GAN (CAEGAN, 2.8), Inverse Conditional Autoencoder GAN (ICAEGAN, 2.9), and Cycle-Conditional Autoencoder GAN (CCAEGAN, 2.10) as incremental cyclical alterations to the conditional GAN (CGAN, (Mirza and Osindero, 2014)) for conditional data synthesis.

Training paradigms comparable to cycles have been proposed in the GAN literature previously (Donahue et al., 2016). Though implemented differently in BiGAN in which the discriminator evaluated corresponding pairs of latent and data points, Donahue et. al. imparted the significance of reconstructing latent codes from samples for the purpose of disentangling learned features (Donahue et al., 2016).

2.8 Conditional Autoencoder-GAN

The CAEGAN (Figure 12) combines reconstructive cycles with a GAN generator by adding a reconstructive loss term to the CGAN. Loss contribution given by discriminator evaluation of generated samples structures the variation space without an explicit prior divergence penalty as in the VAE (Kingma and Welling, 2014).

The autoencoder-GAN collapses the GAN generator and autoencoder decoder into a shared parameter model, where gradients are summed and back-propogated to both the encoder and decoder. The generative and reconstructive tasks share parameterizations, leveraging the assumption that the training distribution and the learned approximation share a latent representation space.

Equation (8) gives the triple-criterion optimized by the CAEGAN:

$$\mathcal{L} = \mathcal{L}_{GAN} + \lambda(\mathcal{L}_C + \mathcal{L}_{recon}) \tag{8}$$

Two of the loss elements are directly borrowed from the nominal models: the adversarial $\mathcal{L}_{GAN}$ (1) and the autoencoding pixel-wise reconstruction loss $\mathcal{L}_{recon}$ given by mean squared error between the input sample and model output. $\mathcal{L}_C$ (Equation 3) is given by a supervised loss from the encoder's prediction of the input sample label, meaning the model must *predict* the corresponding conditions of the input sample in the encoding step.

2.9 Inverse Conditional Autoencoder GAN

The Inverse Conditional Autoencoder GAN (ICAEGAN, Figure 13) serves as the foil to the conditional GAN for an incremental study with the addition of a cycle which recovers latent codes from generated samples. The inverse mapping which returns latent sample estimations of input images not provided by a typical GAN – though it is known to be useful for auxiliary supervised feature learning (Odena et al., 2017). Methods such as contrastive learning (Dai and Lin, 2017) and BiGAN (Donahue et al., 2016) emphasize the importance of recovering the sampled latent code which led to the generation of an image. The reconstruction loss is taken between the latent sample v and the reconstruction $\hat{v}$.

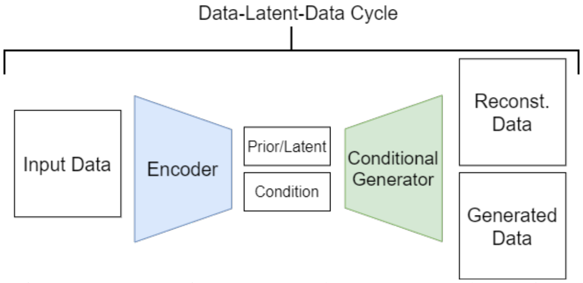


Figure 12: Conditional Autoencoder-GAN (CAEGAN) architecture. Training samples are encoded to a paired latent space given by a variation vector in the prior distribution, and a condition in the label space. From this joint space encoded training samples may be reconstructed, or new samples synthesized from prior sampling.

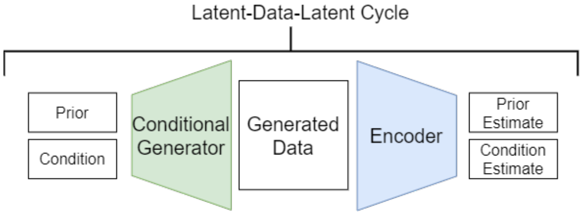


Figure 13: Inverse Conditional Autoencoder GAN (ICAEGAN) Architecture. Pairs sampled from the joint latent-condition space are decoded by a conditional generator, then re-encoded by an encoding function, resulting in a reconstructive loss on the prior and a supervised loss on the condition estimate.

$$\mathcal{L} = \mathcal{L}_{GAN} + \lambda(\mathcal{L}_C + \mathcal{L}_{v\ recon}) \tag{9}$$

The reconstructive loss $\mathcal{L}_{recon}$ is given by recovering the latent sample $\vec{v}$ in the latent variation space V given decoded samples $\mathbf{G}(\vec{v})$. The multivariate standard normal prior and condition space are sampled and decoded by the conditional generator. This decoded random latent sample is evaluated by a discriminator, and each model receives an adversarial loss $\mathcal{L}_{GAN}$ given by (1). Departing from the CGAN, by inducing cycles by reconstructions of encoded or decoded points the generated sample is now re-encoded to the condition-variance space which defines the reconstructive and predictive losses on the recovered variation and condition samples $\mathcal{L}_C$ and $\mathcal{L}_{recon}$ (Equation 3).

2.10 Cycle Conditional Autoencoder-GAN

The Cycle Conditional Autoencoder-GAN (CCAEGAN, 14) is the extension of the training algorithms proposed by the CAEGAN and ICAEGAN. Where the CAEGAN and ICAEGAN perform reconstruction of a space under the image of another space (data-latent-data and latent-data-latent, respectively), the cycle autoencoder-GAN performs both tasks using shared coefficients. This leads to a four part loss, given by:

$$\mathcal{L} = \mathcal{L}_{GAN} + \lambda(\mathcal{L}_C + \mathcal{L}_{recon} + \mathcal{L}_{\vec{v}\ recon}) \tag{10}$$

Where the $\mathcal{L}_C$ now comes from two sources: supervised learning of training data labels $\mathcal{L}_{\vec{c}|\vec{x}}$, and reconstructing the random samples of C-space by generating and re-encoding latent samples from the condition given the decoding of the $(\vec{v}, \vec{c})$

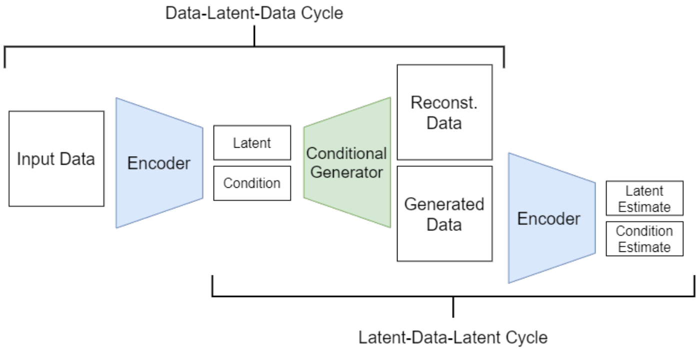


Figure 14: Cycle Conditional Autoencoder-GAN Architecture. Here, two cycles are performed by a shared encoder and generator which incorporate losses from both CAEGAN and ICAEGAN components.

pair:

$$\mathcal{L}_C = \mathcal{L}_{\vec{c}|\vec{x}} + \mathcal{L}_{c|\mathbf{G}(\vec{v},\vec{c})} \tag{11}$$

The CCAEGAN serves as a third incremental step to induce cycles for conditional GANs. The CAEGAN may perform well due to the advantage of reconstructing training samples, which baseline GANs cannot do. This could lead to the model performing well on quantitative metrics by reproducing training samples while failing to produce novel samples (DeVries et al., 2019; Odena et al., 2017). The ICAEGAN and CCAEGAN contrast the CAEGAN as the ICAEGAN only learns from training labels in the same manner as a GAN: indirectly through the lens of the discriminator's feedback.

2.11 Experiments

This section proposes experiments comparing the incremental cyclical models and baselines for generative and supervised learning tasks for a natural image data set. Section 2.12 compares experiments in natural image synthesis quality for the

CIFAR-10 data set quantified by Fréchet Inception and Fréchet Joint Distances, two metrics in image synthesis and conditional image synthesis evaluation (Krizhevsky et al., 2009; DeVries et al., 2019). We include Section 2.14 as a proposed alternative metric for the quality of a trained conditional generator.

Last, we include 2.15 as a proposed direction of research combining generative, supervised, and cyclical models into a united family of distribution-comprehension algorithms which demonstrate adversarial training of natural image synthesis models may improve upon baseline supervised learners.

In order to compare different conditional image synthesis models we emphasize standardized architecture, hyperparameter, training procedures consistent with building blocks from DCGAN (Radford et al., 2015). The DCGAN design is marked by the following characteristics:

- Stride convolutions instead of pooling layers.
- Batch normalization between layers in both generator and discriminator.
- ReLU activations in the generator with a tanh output.
- leakyReLu activation in the discriminator.

Each model used the same architectures and hyperparameters: learning rate $\epsilon = 2e - 4$, Adam optimizer with $\beta_1 = 0.5, \beta_2 = 0.999$, batch size of 16, and 200 training epochs, resulting in $625,000$ training updates per adversary. Each component utilizes five 2-dimensional convolution layers of size 4, stride 2, and padding 1, with ReLU or leakyReLU activations and batchnorm mirroring (Radford et al., 2015).

There exist numerous large GANs which significantly outperform the DCGAN-based models considered here for the FID metric (Mirza and Osindero, 2014; Suman Ravuri, 2019; Brock et al., 2018; Sauer et al., 2022). The models considered here serve as a study on how the minimal addition of cycles to a DCGAN alter FID and FJD outcomes given three possible cycles without changing the generative architecture. Lastly, models such as BigGAN include substantial amounts of training modifications including but not limited to spectral normalization, self attention modules, hinge losses, skip-z connections, orthogonal regularization, and truncation tricks (Brock et al., 2018). Though these changes improve model training stability and FID scores, our goal is to directly compare minimal changes between conditional GANs without these design tweaks.

2.12 Quantifying Generative Quality

This section evaluates the generative quality of the proposed and baseline models using the CIFAR-10 (Krizhevsky et al., 2009) data set, a collection of $60,000$ natural images evenly distributed across 10 content classes. A predefined train-test split is used to evaluate supervised learners on the unseen partition as well as compare distributional distance between generated samples and unseen testing examples to evaluate generative quality using the Fréchet Inception and Fréchet Joint Distances.

The Fréchet Inception Distance (FID, (Heusel et al., 2017)) quantifies the quality of a generated distribution with respect to a target distribution by encoding each in a learned representation in the penultimate layer of the Inception-v3 model, a pretrained natural image classifier (Szegedy et al., 2015). The FID is a standard metric for evaluating generative models (Karras et al., 2020; DeVries et al., 2019; Zhang et al., 2019).

The FID does not account for the joint distribution of samples and classes for

Table 1: Fréchet distances across model architectures for CIFAR-10 synthesis. Lower is better.

	FID	FJD
CGAN	44.37 ± 0.05	51.97 ± 0.05
ACGAN	43.91 ± 0.03	61.59 ± 0.06
CAEGAN (OURS)	37.67 ± 0.37	46.42 ± 0.04
ICAEGAN (OURS)	39.06 ± 0.02	48.16 ± 0.02
CCAEGAN (OURS)	$\mathbf{35.72} \pm 0.02$	$\mathbf{43.22} \pm 0.03$

conditional data to measure class adherence. The Fréchet Joint Distance (FJD, (DeVries et al., 2019)) accounts for joint distributions of images and conditions to express generated sample quality, adherence to the intended class, and distance from other classes. The FJD quantifies the distance between maximum likelihood Gaussian estimation of the conditional distributions in the penultimate layer of Inception-v3.

Table 1 demonstrates the generative quality for baseline and proposed conditional generators. Training and evaluation were performed ten times for each model class. At the conclusion of training, the FID and FJD of the trained model are measured. The variability in the generative quality of these trained models is recorded in the standard error reported for each experiment.

It is worth noting that with the exception of the additional cycle the ICAEGAN and CGAN are identical models, as are the CAEGAN and CCAEGAN. There is no increased model capacity or architectural difference beyond the additional loss components introduced by reconstructive cycles. When quantifying the quality of the generated distribution, the ICAEGAN outperforms the CGAN to a substantial degree. By contrasting these models we demonstrate cycles being a substantial benefit to the training of generative models given a fixed decoder capacity.

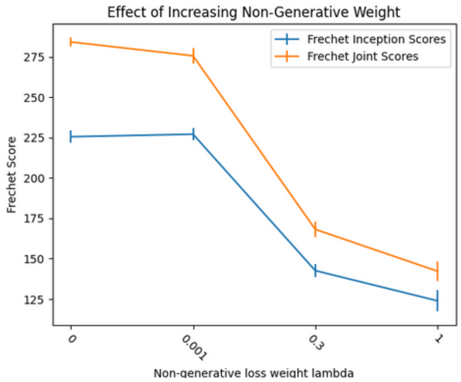


Figure 15: Weighting the non-generative (non-adversarial) losses in a multi-loss model is vital to consistently performant generative models. At 0 weight on the reconstructive and supervised losses, models perform significantly worse than an identical design with weights closer to 1.

The discrepancy between the baseline and CAEGAN models cannot be explained only by the autoencoders generating samples based on leaking reconstruction of the training distribution as described in Section 2.10 as the ICAEGAN does not perform data-space reconstruction of training samples, and the CCAEGAN has diluted the contribution of the data-space reconstruction compared to the CAEGAN. Rather, generative performance is consistently improved when the learned generative weights are updated in part by the addition of non-adversarial cycle losses. Figure 15 demonstrates the relationship of weight on the non-generative losses against the corresponding generative scores as measured by the FID and FJD. Data synthesis outcomes can be improved with non-zero weights on losses not directly relevant to data synthesis.

2.13 Data Mining in a Low-Data Regime

Often, data mining application data sets contain few samples. Shortcomings of adversarial training such as modal collapse and divergence are exacerbated when models (typically the discriminator) overfit to the training data due to the reduced

Table 2: Small-data Fréchet distances across model architectures. Lower is better.

	FID	FJD	DIFFERENCE
CGAN	79.70 ± 0.83	90.56 ± 0.90	10.86
ACGAN	**71.02** ± 0.40	88.58 ± 0.41	17.56
CAEGAN	71.77 ± 0.67	**83.00** ± 0.68	11.23
ICAEGAN	86.04 ± 1.04	99.01 ± 1.19	12.93
CCAEGAN	73.00 ± 0.80	84.11 ± 0.85	11.11

problem difficulty.

In this section we investigate how the performances reported in Table 1 change with a substantially smaller data set. One quarter of the training dataset of CIFAR-10 ($12,500$ images) are used to evaluate potential collapses or differences more apparent in the model designs for this more challenging image synthesis task.

2.14 Augmenting Training Data

The quality of a conditional image synthesizer may be quantified by the down-stream performance of a data mining supervised learner trained using synthesized samples. Synthetic data sets corresponding to a higher classification accuracy may indicate higher quality conditional synthesis, particularly for non-natural image tasks without explicit metrics such as the FID and FJD.

Table 3 reports the down-stream test accuracy of a multi-classification neural network according to a swathe of synthetic sample data set proportions. In each cell, ten trials are performed in which a baseline supervised learner is trained using a training set which is the indicated percentage of the CIFAR-10 training set. The remainder of the $50,000$ images are filled in with synthesized labeled samples given by the row name. This means that for the 75% column, $37,500$ CIFAR-10 samples are chosen from a pre-determined shuffle of the training data constant

Table 3: Test accuracy when training data is augmented with different proportions of generated samples.

Model	75% Real	25% Real	10% Real	5% Real	0% Real
CGAN	69.89 ± 0.05	$\mathbf{64.06} \pm 0.38$	34.13 ± 0.16	24.91 ± 0.09	18.55 ± 0.12
ACGAN	69.45 ± 0.04	44.72 ± 0.20	27.93 ± 0.17	21.24 ± 0.17	16.13 ± 0.15
CAEGAN	69.45 ± 0.04	51.18 ± 0.18	34.45 ± 0.19	25.89 ± 0.15	19.45 ± 0.16
ICAEGAN	69.44 ± 0.05	49.61 ± 0.14	32.78 ± 0.17	25.42 ± 0.13	19.25 ± 0.15
CCAEGAN	$\mathbf{70.00} \pm 0.05$	52.41 ± 0.19	$\mathbf{36.49} \pm 0.21$	$\mathbf{26.46} \pm 0.10$	$\mathbf{19.95} \pm 0.10$

for each experiment, and the image of $12,500$ latent samples are drawn from the corresponding generator and concatenated to the data set.

Results in Table 3 demonstrate how using conditional sample synthesis contributes to model testing performance for a variety of ratios of training data to GAN-augmented data, measuring the trained generator's adherence to semantic content present in CIFAR-10 testing samples as determined by the testing loss of a model trained on the conditional synthesized samples.

2.15 Cyclical Models Perform Classification

The final result on the benchmarking contribution of research is the demonstration that cyclical models do not just improve on comparable CGANs for image synthesis, but situationally outperform comparable supervised learners on image classification. One motivation for the study of generative models is improved utilization of training data for robust, efficient models. Efficiency of training data becomes increasingly vital for deep learning success as the number of samples decreases. The encoding component of each of the cycle models CAEGAN, ICAEGAN, and CCAEGAN perform the multiple attention task of encoding samples to the variation and code spaces. To measure supervised learning outcomes each cell of Table

Table 4: Top-1 test accuracy of classification heads of conditional cycle models. Higher is better.

Model	75% of Train	25% of Train	10% of Train	5% of Train
Predictor	**69.87** ± 0.23	**59.12** ± 0.06	**47.09**± 0.07	32.60 ± 0.12
CAEGAN	68.55 ± 0.04	56.34 ± 0.08	45.78 ± 0.07	**41.07** ± 0.17
ICAEGAN	18.60 ± 0.80	20.71 ± 0.56	15.30 ± 0.89	14.05 ± 0.41
CCAEGAN	65.25 ± 0.25	50.86 ± 0.25	41.21 ± 0.18	34.30 ± 0.13

4 indicates ten models trained from scratch on a CIFAR-10 subset.

Instead of training only a predictive model, the three cycle models each use the encoding portions of their architectures to encode the testing samples alongside adversarial generative and reconstructive learning.

Table 4 demonstrates that the complexity of handling autoencoding, generative, and supervised tasks is a burden for the cyclical models for large volumes of data. Though the CAEGAN and CCAEGAN testing accuracy is in stride with the simple predictor for 75% and 25% of the training set, the results fall off for the CCAEGAN when using 10% $(5,000)$ of training samples. However, a turning point exists between 10% ($5,000$ samples) and 5% ($2,500$ samples) of the original set: the regularization imparted upon the encoder by managing the generative, reconstructive, and predictive tripartite loss improves the model's testing accuracy, as both the CAEGAN and CCAEGAN pull ahead of the comparable supervised learner. The ICAEGAN performance remains poor throughout due to the encoder training without direct access to the training data set.

2.16 Chapter Conclusions and Subsequent Directions

We have demonstrated how the inclusion of cycles improves upon comparable CGANs. We demonstrate that the relationship between conditional image synthesis and supervised learning may benefit both tasks as performance in one area may be used as a metric for the other. These results were consistent across the FID, FJD and proposed augmentation metrics for CIFAR-10. The consistency with which the three highly different cyclical models outperformed the baselines lends itself to the notion that alternative training metrics may regularize against shortcomings of adversarial training by limiting the space of viable parameters.

Generalizations on the theme of cycles for reconstruction may be extended to any model architectures which rely on learning representations of data, and may be beneficial beyond conditional GANs for image synthesis including extensions to autoregressive and supervised learners. Contemporary research including (Karras et al., 2020) investigates the utilization of augmentations for stabilizing GAN training, particularly for limited-data domains. The study of regularization effects including but not limited to the corresponding quality of synthesized images and model inference during training may be a productive area of research for the stabilization of GAN training. Despite notorious instability in GAN training particularly for low-n tasks, concurrent training of generative and supervised models may be a promising direction for data-efficient multi-task deep learning models.

Subsequent chapters will call upon the novel architectures and multitask training as presented here. Chapter 3 will utilize alternative multitask training paradigms inspired by the meaningful latents found in 2.7. Chapter 4 uses strongly-structured latents as an alternative to the weakly-structured latents found in Chapter 2. Research from this Chapter was extended using adversarial training with CAEGAN-ACGANs and time series chemical sensing data in a forthcoming paper *Multitask Conditional Data Synthesis with Cycles*.

3 Multitask Adversarial Training

Chapter 1.3 motivated the development of a chemical sensor capable of discrimination of a broad range of chemical analytes. From the hardware perspective, an array of semi-selective chemical sensors may respond to many analytes simultaneously. The machine learning component of the device, however, may fail to learn decision boundaries which remain accurate in the detection of obscured or novel chemical analytes with a traditional classifier.

In order to measure model response to a likely change in real-world deployment distribution caused by obscurant analytes, we utilize single-analyte exposures of four chemical analytes of interest (Analyte A, B, C, or D). Models trained with these single-analyte exposures are tested on unseen double-analyte exposures (combinations A-B, A-C, A-D, B-C, C-D) in which Analyte A may be present but masked by some unknown obscurant analyte. Section 3.2 discusses the complications of gathering exhaustive experimentation beyond single-analyte exposures. The classification of Analyte A in the presence of obscurant analytes emulates the vital task of hazardous analyte detection for real-world applications where environmental factors are unpredictable for experimentation.

We propose the utilization of Auxiliary Classifier Generative Adversarial Networks (ACGANs) as supervised learners which make use of adversarial and multitask training to improve out-of-distribution performance (Odena et al., 2017). Though generative deep learning has been applied to molecular synthesis and drug discovery (Elton et al., 2019; Gebauer et al., 2022), the inclusion of adversarial and generative training to chemical analyte discrimination may be beneficial to chemical discrimination with sensor devices and to our knowledge are yet unexplored (Ruder, 2017; Mater and Coote, 2019; Wei et al., 2019). We hypothesize that simultaneous adversarial training of a data-synthesizing model improves generalizability of the discriminator-classifier model through the parameterization bias imparted by multitask learning (Caruana, 1997; Shinohara, 2016; Child, 2021). Results in Section 3.5 demonstrate inducing additional data-synthesizing tasks (Section 3.3) without additionally annotated data significantly improves testing outcomes for distribution-shifted data (Section 3.2).

3.1 Chapter Contributions

- We propose the utilization of ACGANs as generalizable multi-task learners which improve testing classification outcomes on challenging distribution shifts. We leverage multi-task learning without requiring additional experimental labels or annotations (Section 3.3).

- We record optimized benchmarks for four baseline machine learning models as well as a comparison between feedforward neural network classifiers and ACGANs given optimized hyperparameters for discrimination of Analyte A as well as detection in the presence of obscurant analytes (Section 3.4, 3.5).

- We perform model comparisons between comparable ACGAN and feedforward neural networks to diagnose how inducing additional tasks may benefit generalizability of deep learners for chemical sensing under distribution shifts (Section 3.5).

- We study the scaling behavior of classifier models with respect to the combination of single- and double-analyte training samples to improve future chemical sensor data set development (Section 3.6).

3.2 Data

The data considered throughout this section are drawn from two types. Single-analyte exposures are created by exposing one chemical analyte of a given concentration to the multi-sensor array for a constant length exposure. The second, more challenging data are the double-analyte exposures: a two-analyte mixture is exposed to the sensor array for a time period. These double-analyte exposures are substantially more difficult to classify as the desired information (the presence of Analyte A) is obscured by an additional analyte which alters the sensor response profiles (Figure 8). The introduction of three other obscurant analytes each modifies the resistance curves in different ways. Though Analyte A is still present, the relationships of sensor responses from the unobscured curve have been altered. In order to measure model quality for a challenging obscured-analyte out of distribution task, a supervised model must account for this change in distribution without access to these double-analyte responses during training.

In Figure 17, we see separation between samples containing Analyte A and those not containing Analyte A. Experiments One and Two may have sensor coatings more conducive to the discrimination of Analyte A from other analytes as judged by the distance and linear separability and distance of Analyte A samples under one ISOMAP transformation (Tenenbaum et al., 2000).

For chemiresistive sensor signal classification, an example machine learning approach to the discrimination of chemical Analyte A from chemical Analytes B, C, and D succeeds when one analyte is present at a time (Figure 18a). For the more complex detection test of obscured signals given by analyte mixtures, the generalization performance of this model is poor, as the positive class samples are distributed along both sides of the decision boundary (Figure 18b). The ability to

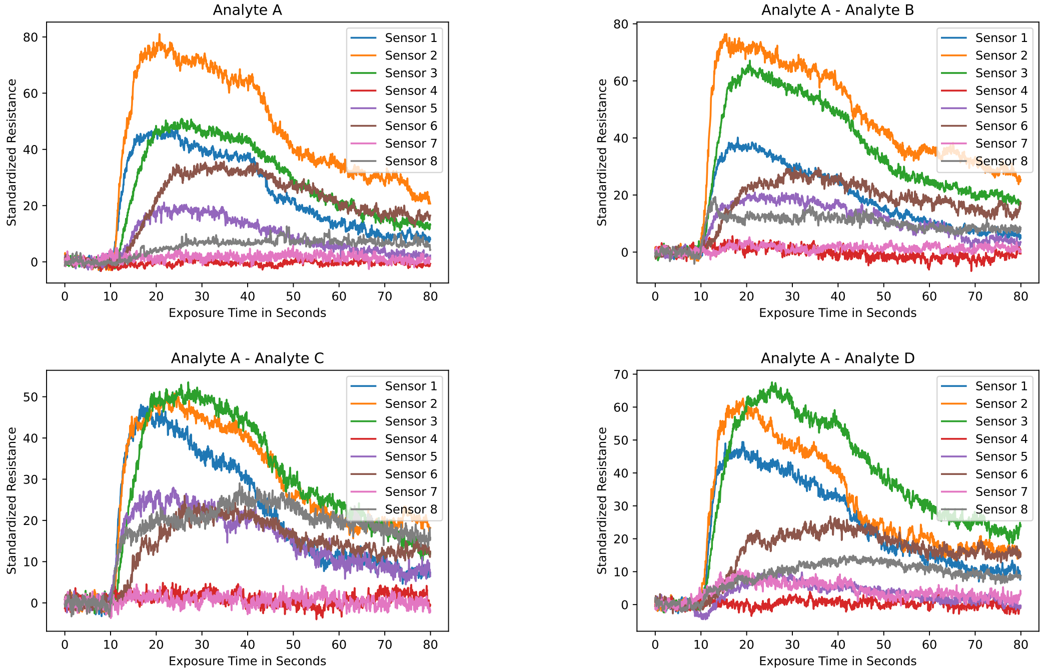


Figure 16: Sensor response curves to Analyte A are shown unobscured (top left) and obscured in three different ways. Obscurants change the characteristic sensor resistance response of Analyte A alone.

discriminate Analyte A signals from other analyte exposures even when obscured by a secondary analyte is a vital task for chemical sensors, and one which demands changes to standard data mining practice.

In addition to being harder to classify, mixture data are substantially harder to exhaustively gather. The experimental cost of datasets even with limited numbers of analytes and concentrations quickly becomes intractable as trials require expert supervision, experimental design, hardware setup, and data verification processes. Given a number of chemical analytes of interest $x_1, ..., x_n$ at various concentrations $c_1, ..., c_k$, the number of experiments necessary to perform experimentation of all mixtures at all concentrations is given by

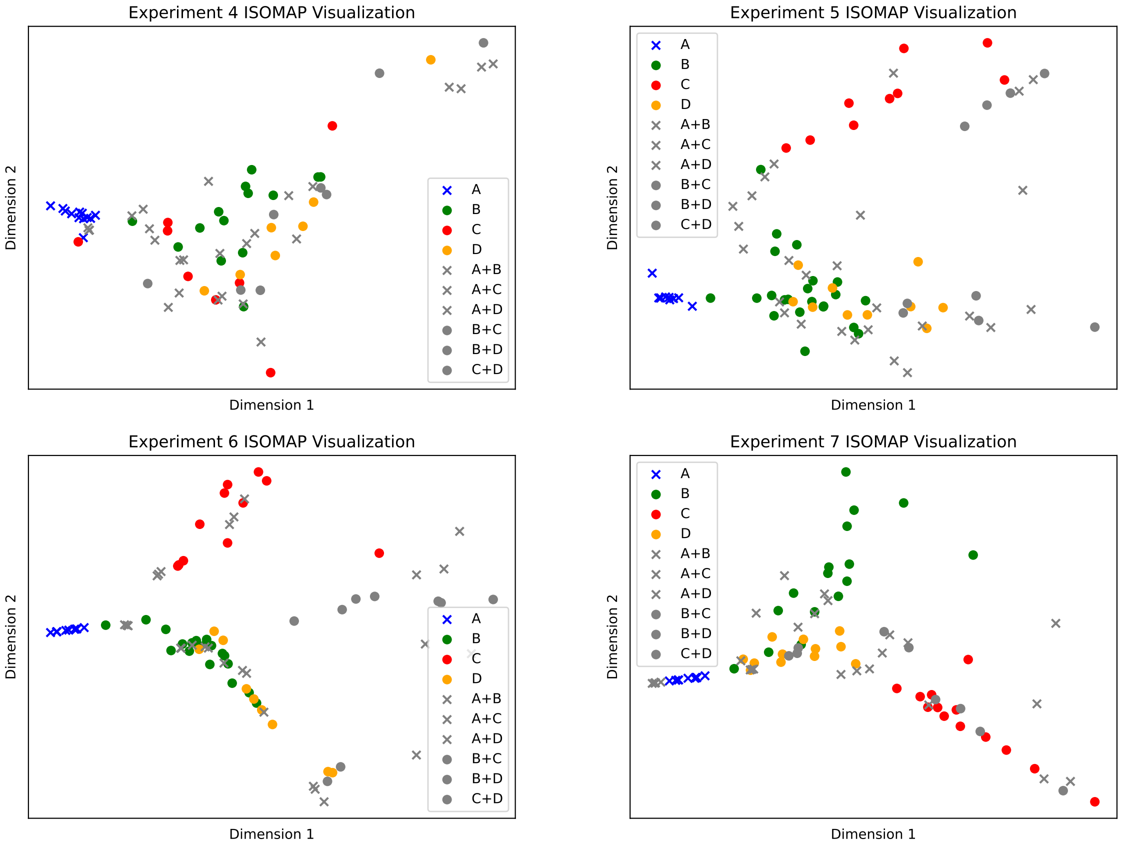
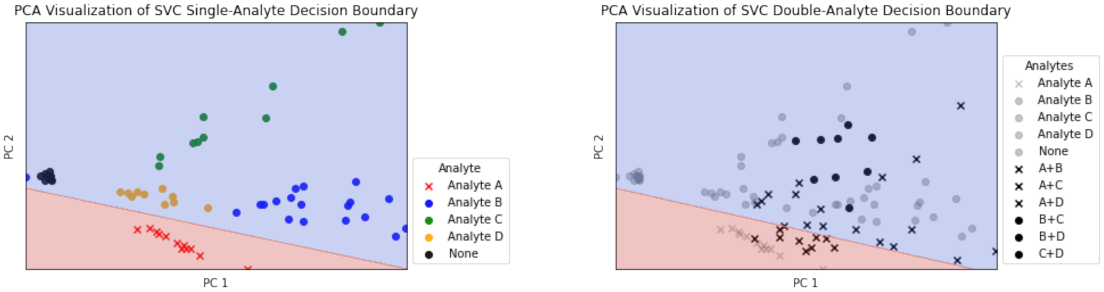


Figure 17: Exploratory data analysis demonstrating ISOMAP dimensionality reduction (Tenenbaum et al., 2000) from the space of sensor response curves to two dimensions for four sensor array data sets. In this visualization, an ISOMAP embedding trained on single-analyte exposures then encodes both the single and double-analyte sets. Samples containing the analyte of interest Analyte A are marked with crosses to differentiate the positive classification label. We find double-analyte exposures distributed among single-analyte exposures, and positive-class samples distributed among negative classes.



(a)

(b)

Figure 18: Visualization of the shortcomings of linear decomposition with linear classifiers for mixture data. **Left**: Under a 2-dimensional PCA transformation, signal data of single-analyte exposures may be well-separated by a linear classifier. **Right**: Under a PCA embedding of signal data, the same model fails to accurately discriminate between samples containing Analyte A. Samples marked with x contain Analyte A and are intermixed with non-Analyte A single-analyte exposures, leading to poor classification results from this linear classifier.

$$\sum_{p=1}^{n} \binom{n}{p} k^{p} \tag{12}$$

where p represents the number of analytes in the mixture. Figure 19 visualizes the complexity of gathering representative data sets for even a limited number of analytes and concentrations: in this case, simultaneous exposure of one, two, three, or four chemical analytes each at one of four potential vapor concentrations.

Limitations in experimental data volume lead to investigations in improving model performance on a challenging chemical sensor problem - particularly when the space of possible analyte exposures is highly complex and expensive to sample from. Generalizing from laboratory to real-world data incurs additional complexity due to distributional differences which may lead to detection failures in machine learning tools (Bousmalis and Levine, 2017). In classifying the presence of Analyte A borne in vapor by an olfactory system, it is vital to measure the performance of the classifier for unseen obscurant analytes. Since it is prohibitive to

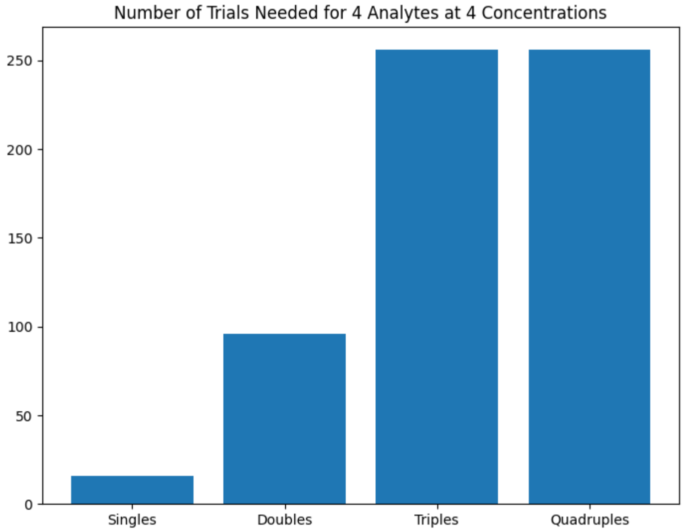


Figure 19: The corresponding number of experimental trials to be performed given four analytes (A, B, C, D) at four possible exposure magnitudes (12.5%, 15%, 20%, 25%).

perform exhaustive experimentation of double-analytes in a laboratory setting, we turn to the optimization of classifiers in order to handle unseen challenging distributions.

Triplicate experiments are performed for each of four analytes at each of four concentrations$\{6.25\%, 12.5\%, 17.5\%, 25\%\}$. We retain an unseen holdout set of double-analyte exposures. This small data set ($n \approx 30$ for each of three experimental data sets) of two simultaneous analyte exposures are used as the holdout set to measure how well methods are able to generalize to unseen distributions of obscured analytes.

3.3 Generative Multitask Models

ACGANs are one approach to induce a multitask training paradigm with no additional labels or exterior tasks. Rather than including additional tasks with extra labels on training samples, the model utilizes additional *induced* tasks via the

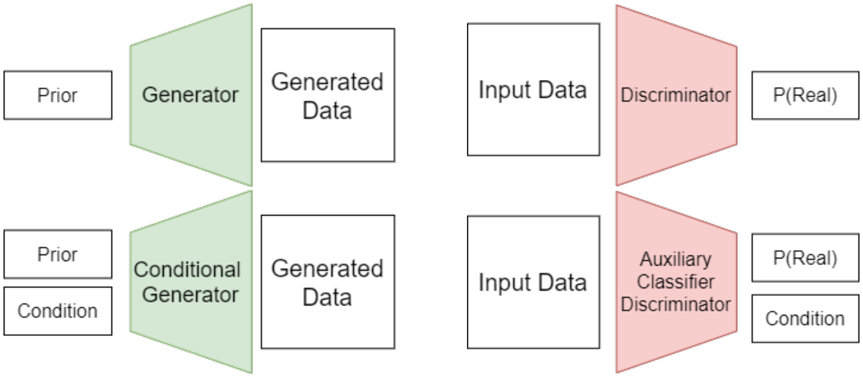


Figure 20: The ACGAN modifies the GAN architecture to include an auxiliary classifier head on the discriminator output layer. A condition classification task extends a data synthesizing model to a conditional data synthesizing model. We utilize the classifier output which shares parameter optimization with the discriminator as a regularized supervised learner.

adversarial data synthesis process which optimize the model with respect to the same training data and labels but with additional loss terms (Caruana, 1997). We hypothesize the representation bias phenomena (Baxter, 2000) from the additional tasks improves model parameterizations. Deeper representations may improve supervised learning generalizability over comparable feedforward classifiers, particularly for low-data experiments with a shifted testing distribution.

The Auxiliary Classifier GAN (ACGAN, interested readers may refer to (Odena et al., 2017) on adversarial supervised training) modifies the discriminator of the GAN to predict the corresponding class of input training and synthesized samples. ACGANs account for conditional probabilities approximating the joint data distribution $p(\vec{x}, \vec{c})$ by conditioning the generation and discrimination of samples on the analyte label $\vec{c}$. In training, the generator **G** uses a condition $\vec{c}$, here representing a chemical exposure label, to conditionally generate from a random latent sample z converging to an approximation of the conditional training distribution. The ACGAN alters GAN training by incorporating an auxiliary classifier head on the output of the discriminator, with a conditional component given by:

$$\begin{aligned} L_C &= \mathbb{E}_{\mathbb{X}}[\log P(C = c|\vec{x})] \\ &\quad + \mathbb{E}_{\vec{z}}[\log P(C = c|\mathbf{G}(\vec{z}))]. \end{aligned} \tag{13}$$

Using the conditional loss, the Discriminator is trained to maximize $L_C + L_{GAN}$ (1) while the Generator is trained to maximize $L_C - L_{GAN}$ (Odena et al., 2017). Odena (2017) finds the inclusion of the conditional adversarial training quantifiably stabilizes training and improves discriminability of generated samples alongside advantages in representation bias through multitask learning by balancing adversarial and supervised losses (Caruana, 1997). Utilizing the conditional discrimination of the ACGAN to classify samples from a shifted testing distribution with results from Section 3.5 demonstrates that adversarial training improves upon feedforward networks for three distribution-shifted chemical sensing data sets.

3.4 Methods

The data are split into two categories for the experiments in Section 3.5. Single-analyte exposures compose the training set which is used to train machine and deep learning classifiers to predict the presence of Analyte A. The single-analyte exposure data are also used for validation in which unseen holdout samples are used to optimize over hyperparameter grid searches (Algorithm 1) in Section 3.4.

Results in Section 3.5 use three separate real chemical sensing data sets with different sensor materials, analyte concentrations, and analyte exposures. Each experiment is separated into a single-analyte data set and a double-analyte data set. Single-analyte data sets are utilized as training and validation data for hyperparameter optimization. Double-analyte exposure experiments are utilized as testing data.

For each set of experiments, we train the following supervised learners which represent a diverse set of baseline classifiers. Each model is optimized over a broad swathe of hyperparameters selected by cross-validation given the experiment, data set, and exposure time period $t \in \{1, 1.25, 1.5, 1.75, 2, 2.5, 3, 4, 5\}$ seconds. The hyperparameter grid searches utilized for each model are as follows:

1. Auxiliary Classifier GANs (Odena et al., 2017)
    - Model width: 8, 16, 32, 64, 128
    - Learning rate: 0.001, 0.002, 0.01, 0.05
    - Training epochs: 4, 8, 16, 32, 64, 128
    - Batch size: 4, 8, 16
    - Latent Dimension: 32
2. Feedforward Neural Networks (Rumelhart et al., 1986)
    - Model width: 8, 16, 32, 64, 128
    - Learning rate: 0.001, 0.002, 0.01, 0.05
    - Training epochs: 4, 8, 16, 32, 64, 128
    - Batch size: 4, 8, 16
3. Baseline models (Implemented via (Pedregosa et al., 2011)):
    (a) Decision Tree Classifier (Breiman et al., 2017)
        - Max depth: 1-8
        - Minimum samples per split: 2-8
        - Max features considered: 1, 2, 4, 8, $\sqrt{vars}$

(b) Random Forest Classifier (Ho, 1995)

- Number of estimators: 2, 5, 25, 100, 200, 400, 800
- Splitting criterion: gini, entropy
- Max depth: 2, 4, 8, 16, 32

(c) K-Neighbors Classifier (Silverman and Jones, 1989)

- Number of neighbors: 1-20
- Weights: Uniform weighting, distance weighting
- Distance power: 1, 2

(d) Logistic Regression (Cox, 1958)

- Parameter Lasso penalty: 0.0001, 0.001, 0.01, 0.05, 0.1, 0.15, 0.25, 0.5, 0.75, 1.0

For each of the six models the following optimization method is used. For each experiment and for each time $t \in \{1, 1.25, 1.5, 1.75, 2, 2.5, 3, 4, 5\}$ seconds, the model is trained on half of the training data using a stratified sampling by analyte concentration. The other half is the validation set used to select the optimal hyperparameter configurations given 5-fold cross validation (Stone, 1974). The optimal model from the exhaustive hyperparameter search is taken to be the highest-performing model based on validation set F1 (Pedregosa et al., 2011) score for that time period. The F1 score is given by the harmonic mean of the precision and recall of the model classifications. The validation process is repeated 5 times to find optimal hyperparameters, and the corresponding testing F1 score is recorded. The mean and standard error of testing performance is provided for each data set, experiment, and model in Section 3.5.

Algorithm 1 A hyperparameter gridsearch algorithm for one model, experiment, and time period.

```
Require: Supervised learning model $m$. Exposure time $t$. A number $k$ of folds for cross-validation. A set of hyperparameters to consider $\{h_0, ..., h_p\}$. A number of trials to re-perform the experiment for standard error. Training single-analyte dataset ($n \approx 100$) from one of three experiments. Testing double-analyte dataset ($n \approx 30$).
  Declare a split $K$ of the training data into $k$ disjoint partitions using label-stratified sampling.
  for each trial do
      for $h \in H$ do
          for each train-validation split $i \in K$ do
              Remove $i^{th}$ partition of training data as validation.
              Train a classifier on the training partition using the hyperparameter configuration.
              Record the model's F1 score on the validation set.
          end for
      end for
      Select the model hyperparameter configuration with the lowest mean validation score.
      Retrain this model on the entire training set.
      Record this model's testing F1 performance.
  end for
  Summarize mean and standard error of testing F1 performances across trials given optimal models.
```

Table 5: Mean ± standard error testing F1 score of classification of Analyte A given single-analyte training and testing sets. Models are cross-validated for hyperparameter selection on a validation set before being tested on a holdout set of single-analytes. For each experimental data set, the best overall model is highlighted in dark green and the second best overall model is highlighted in light green.

Model	Experiment One	Experiment Two	Experiment Three
ACGAN	0.7540 ± 0.0121	$\mathbf{0.7968} \pm 0.0114$	0.8684 ± 0.0130
Neural Network	$\mathbf{0.8504} \pm 0.0113$	0.6980 ± 0.0287	0.6186 ± 0.0140
Decision Tree Classifier	0.2679 ± 0.0536	0.4075 ± 0.0815	0.7529 ± 0.1506
K-Neighbors Classifier	0.4254 ± 0.0851	0.5271 ± 0.1054	0.8448 ± 0.1690
Logistic Regression	0.3661 ± 0.0732	0.4610 ± 0.0922	$\mathbf{0.8843} \pm 0.1769$
Random Forest Classifier	0.2969 ± 0.0594	0.4875 ± 0.0975	0.7949 ± 0.1590

3.5 Results

We report the improvements to rapid and accurate classification of an analyte of interest in the presence of potential obscurant analytes. Throughout, the F1 score is used as the metric to quantify success.

Results from Table 5 indicate a lack of certainty in the superior model for the classification of Analyte A without the presence of obscurants in the testing set. Multiple models perform well on a single-analyte training set when optimized on single-analyte training data. ACGANs perform best on Experiment Two single-analyte classification data, and are second best for Experiments One and Three.

Results from Table 6 indicate that the usage of the ACGAN discriminator as a supervised learner dramatically improves testing outcomes across data sets compared to comparable neural networks and baseline machine learning models. Changing the testing distribution significantly reduces F1 metric outcomes compared to results in Table 5, but the ACGAN loses less ground when compared to peer models which demonstrate a failure to generalize to the testing set.

Figure 21 visualizes the difference between comparable feedforward neural network and ACGAN models for a change in testing distribution for all three

Table 6: Mean ± standard error testing F1 score of classification of Analyte A in the presence of an additional analyte given training on single-analyte exposures. Models are cross-validated for hyperparameter selection on a validation set of single-analytes before being tested on an unseen set of double-analyte exposures to measure the generalizability of classifiers to new distributions. For each experimental data set, the best overall model is highlighted in dark green and the second best overall model is highlighted in light green.

Model	Experiment One	Experiment Two	Experiment Three
ACGAN	**0.4436** ± 0.0493	**0.5525** ± 0.0614	**0.4655** ± 0.0517
Neural Network	0.1804 ± 0.0497	0.1804 ± 0.0497	0.0691 ± 0.0402
Decision Tree Classifier	0.2169 ± 0.0434	0.2098 ± 0.0420	0.1119 ± 0.0224
K-Neighbors Classifier	0.2755 ± 0.0551	0.2287 ± 0.0446	0.0906 ± 0.0181
Logistic Regression	0.2339 ± 0.0468	0.2231 ± 0.0446	0.1170 ± 0.0234
Random Forest Classifier	0.1474 ± 0.0295	0.1797 ± 0.0359	0.0111 ± 0.0022

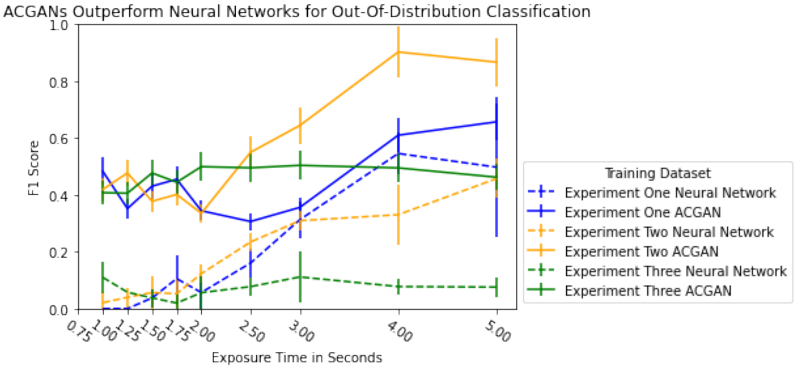


Figure 21: Across data sets and exposure times, ACGANs trained with single-analyte exposures outperform comparable feedforward neural networks when tested on double-analyte exposures. These performances account for a cross-validation optimized model over a large gridsearch of parameters for each possible exposure window and experimental data set.

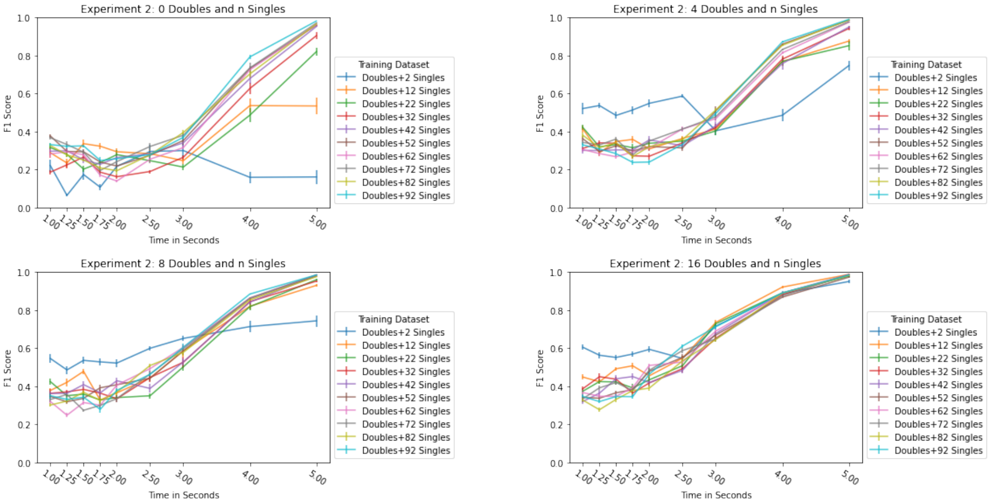


Figure 22: For Experiment Two, we demonstrate F1 score outcomes for the classification of double-analyte samples for variable time windows. For each combination of a random sample of k double-analyte experiments and n single-analyte experiments, the resulting F1 outcome on holdout double-analytes is reported for each exposure window time. Moving from top left to bottom right, the classifier is trained with incrementally more double-analyte exposures.

data sets. Though optimized on comparable hyperparameter sets, the addition of the adversarial data synthesis task for ACGANs has substantially improved the performance on the more challenging double-analyte distribution for all time windows. Though F1 improvement trends of neural networks and ACGANs are comparable when increasing exposure time, ACGANs perform substantially better on generalization to an out-of-distribution testing set of obscured double-analytes across all exposure windows with the exception of a near-time for three- and four-second exposures in Experiment One.

3.6 Scaling Behavior

Figure 22 explores the possibility of including double-analyte training data to improve learning outcomes for double-analyte classification. As the number of double-analyte experiments included in training increases, we find decreased variability in F1 outcomes with respect to variable numbers of single-analyte training data. In addition, including 16 double-analyte samples significantly improves learning outcomes for all numbers of additional single-analytes. There exists an inflection point between 0 and 4 double-analyte samples which causes the addition of further single-analyte samples to detract from training, particularly in small time window detection problems. The variance in classification outcomes is highly dependent on time and number of single-analytes for zero double-analytes (top-left), and consistently low with 16 double-analyte training samples, regardless of the number of singles-analyte samples.

The addition of double-analyte training samples improves classification outcomes for these chemical sensing data sets. In addition, we have demonstrated little improvement beyond 32 single-analyte training samples on the learning outcomes for double-analyte classification. These two results together indicate that given the experimental design used in this paper with four analytes. further experiments may be required to sample from the space of double-analyte exposures. Though this effect is consistent in three chemical sensing experiments using four analytes of interest, it may not hold for higher numbers of analytes as the scaling behavior of experimentation with high numbers of analytes. We may expect similar scaling behavior in the number of double-analyte experiments used as training data for differing analytes, analyte combinations, or sensor designs. However, exhaustively sampling from double-analytes rises quadratically with the number of analytes of interest, whereas single-analyte sampling used in this research remains linear.

3.7 Discussion

Though GANs are known to suffer collapse in low-data training, we demonstrate that the ACGAN may be trained such that its supervised generalizability is significantly improved beyond that of comparable neural networks or baseline models. This finding may be corroborated by ongoing research into the representations learned by multitask synthesizing-predicting models and investigations into the apparent data efficiency and robustness to distribution shifts of these adversarial-trained models. It is possible that deeper, more generalizable representations are learned as a function of additional optimization terms induced from data synthesis.

Data synthesizing models represent a highly flexible function from a latent prior to a complex manifold approximating the training distribution. For this reason further analysis of the role of generators as adaptive augmentations for the training of classifying discriminators is necessary. Auxiliary classifier discrimination evaluation of synthetic samples may serve as a form of training data augmentation factoring into the increased generalizability seen in Section 3.5. Conditional generators could play a similar role to augmentations in the training of auxiliary discriminators, and further research is needed into the relationship between augmenting data sets with synthetic samples versus perturbed ones particularly in the adversarial learning paradigm at various training data volumes and complexity.

For sensor tasks where the testing distribution is known to depart from training, adversarial-trained classifier models may be appropriate for tasks in which gathering exhaustive data on the new distribution may be intractable. The representation bias from multitask models may benefit supervised learners for low-data paradigms, even in the absence of additionally-annotated data. Further research will extend results from this chapter to incorporate cycles for multitask generation as in Chapter 2. Incorporating time series neural network architectures into the ACGAN components discussed here will further improve results from a supervised and unsupervised perspective on chemical sensor classification.

4 Zero-Shot Learning with Semantic Training

Chapter 1 motivated accurate chemical sensors as vital to industry and safety applications. Training machine learning models to be accurate on real world chemical sensor data requires performing many diverse, costly experiments in controlled laboratory settings to create a data set. In practice even expensive, large data sets may be insufficient for generalization of a trained model to a real-world testing distribution. Rather than perform greater numbers of experiments requiring exhaustive mixtures of chemical analytes, this chapter proposes learning approximations of complex exposures from training sets of simple ones by using single-analyte exposure signals as building blocks of a multiple-analyte space. We demonstrate this approach to synthetic sensor responses surprisingly improves the detection of out-of-distribution obscured chemical analytes. Further, we pair these synthetic signals to targets in an information-dense representation space utilizing a large corpus of chemistry knowledge. Through utilization of a semantically meaningful analyte representation spaces along with synthetic targets we achieve rapid analyte classification in the presence of obscurants without corresponding obscured-analyte training data.

Transfer learning for supervised learning with molecular representations makes assumptions about the input data. Instead, we borrow from the natural language and natural image processing literature for a novel approach to chemical sensor signal classification using molecular semantics for arbitrary chemical sensor hardware designs. The techniques for semantic training introduced here have been demonstrated in the natural image and natural language literature to improve accuracy and reduce catastrophic errors in classification. Incorporating these techniques may help chemiresistive sensor arrays classify challenging distributions

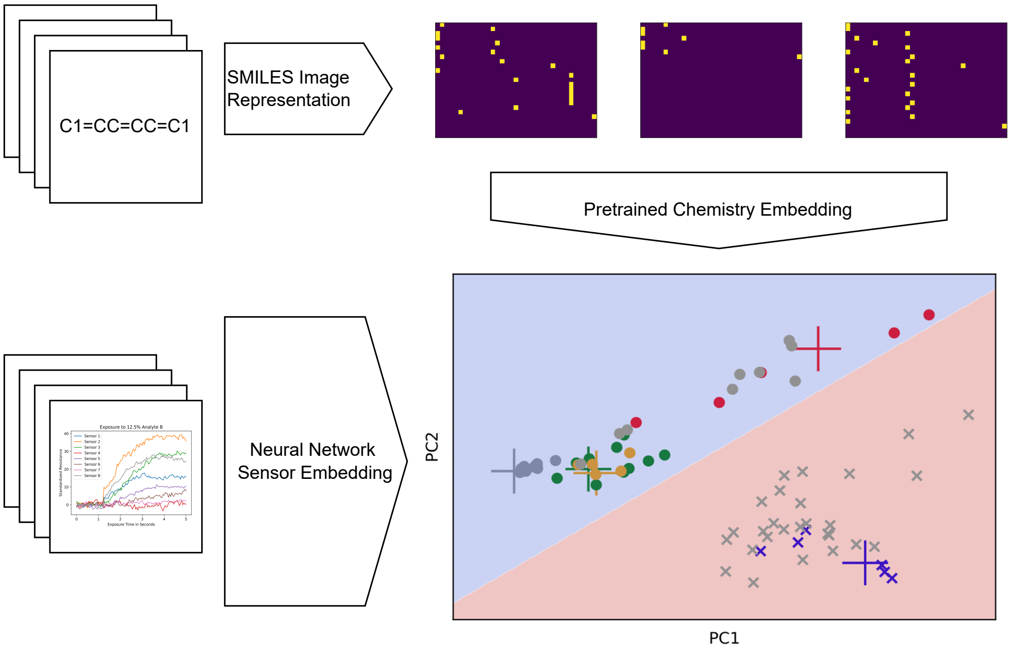

C1=CC=CC=C1
SMILES Image Representation
Pretrained Chemistry Embedding
Neural Network Sensor Embedding
PC2
PC1

Figure 23: ChemVise utilizes activations from a pretrained supervised molecular property model to represent analytes of interest with geometric interpretation. The utilization of arithmetic means between points in analyte space creates improved, tunable decision boundaries for chemical sensor classifiers. We find that semantic chemistry representations outperforms non-chemical embeddings and leads to greater discriminability of obscured chemical analytes.

and raise the floor on out-of-distribution performance.

4.1 Introduction

Classifying chemical vapors is vital to military, industrial, and safety applications. Hazardous chemical detection becomes challenging in the presence of obscurants and environmental factors as sensor responses change. This chapter addresses the generalizability of data mining applied to challenging chemical vapor mixtures with novel deep learning approaches. We take inspiration from embedding-translation models in calling our approach "ChemVise", a portmanteau of the constituent Chemception (Goh et al., 2017) molecular attribute prediction model and

DeViSE (Frome et al., 2013) visual-semantic embeddings. Rather than natural image and sentence semantics, we utilize deep representations of SMILES molecular images paired with chemiresistive sensor signals to learn improved representations for downstream predictors (Figure 23). Improved chemical representations for chemical sensing allows even challenging double-analyte samples to be encoded such that they are linearly separable with simple classifiers. We find our approach significantly outperforms competitive models for the detection of obscured chemical analytes with only unobscured analytes as training data.

In order to classify more challenging analyte mixtures which do not appear in the training set, we borrow from zero-shot, transfer, and multi-modal learning approaches in deep natural image and language learning (Larochelle et al., 2008; Frome et al., 2013; Radford et al., 2021). We contribute the following to the challenging task of obscured analyte detection for one chemical detection paradigm, with generalizations available for any sensing hardware design:

1. We introduce the novel ChemVise approach which modifies transfer learning to utilize chemistry domain knowledge for any chemiresistive sensor data (Section 4.2).

2. We apply linear combinations for analyte mixture molecular-semantic representations for the improved detection of obscured analytes (Section 4.3). The utilization of linear combinations for chemistry representations as well as linear combinations for sensor response to mixture data is yet unexplored in the literature.

3. We demonstrate that transfer learning of ChemVise outperforms other machine and deep learning approaches to chemical sensing as well as alternative non molecular-semantic representations (Section 4.4).

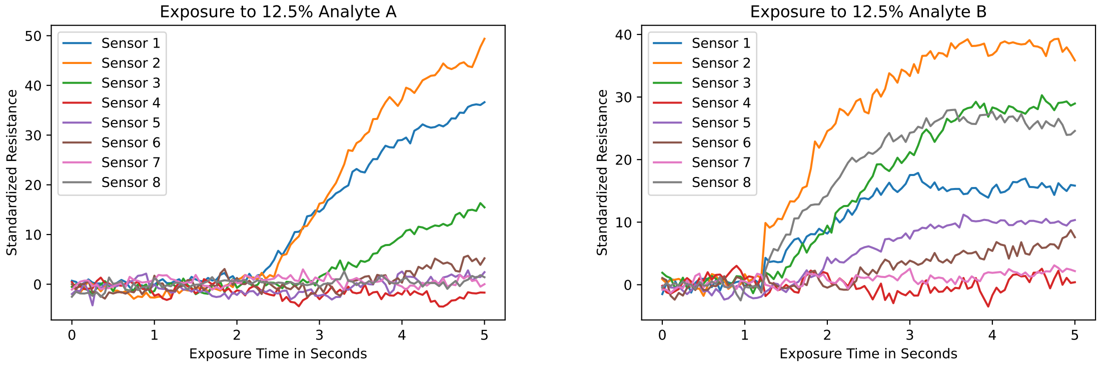


Figure 24: Two example single-analyte exposures. **Left**: A 5-second window including the moment a mixture of 12.5% Analyte A is exposed to the array of eight sensors. **Right**: A 5-second window including the moment a mixture of 12.5% Analyte B is exposed to the array of eight sensors. Machine learning classifiers use discrepancies in sensor resistances responses to discriminate between analytes.

4. We provide an outline for the generalization of this approach to arbitrary sensing hardware. Whereas most transfer learning techniques would be domain specific, this approach can adapt the training of the deep encoder for arbitrary input data to the same latent representation space (Section 4.7).

The ability to import significant domain knowledge in the form of transfer learning is vital to the rapid development of diverse and accurate sensors. Promising results in chemistry-applied deep learning indicate that well-tuned models shorten the development cycle of new tools and in some cases can replace repetitive laboratory experiments (Wei et al., 2019).

By utilizing single-analyte exposures canonical dimensions of multi-analyte mixture space and incorporating external chemistry knowledge, supervised learning outcomes are improved and constraints to experimental hardware, budget, and time limit may be mitigated.

Our surveying efforts above report many other techniques utilizing molecular and compound embeddings for downstream supervised learning. However, to our knowledge our ChemVise approach (Section 4.2) utilizing transfer learning representations as targets to train embedding models for arbitrary input data remains unexplored in chemical sensing and materials science. Natural image and natural language researchers (Frome et al., 2013; Socher et al., 2013; Radford et al., 2021) have emphasized semantically meaningful target spaces for supervised and generative deep learning. Sentence-to-image generation and image-to-sentence captioning are a promising recent breakthrough which large language transformers have had incredible recent successes (Ramesh et al., 2022). Multi-modal deep representation learners use paired encoder-decoder models which operate in the same meaningful embedding space, and learn to link the two embeddings with a translation model. Rather than natural image and sentence semantics, however, our ChemVise approach utilizes deep representations of molecular SMILES images encodings paired with chemiresistive sensor signals to learn improved latent representations for downstream predictors (Figure 25).

4.2 ChemVise Method

Our proposed ChemVise approach to chemical sensing borrows from the deep learning literature on zero-shot learning (Larochelle et al., 2008), transfer learning (Weiss et al., 2016), and domain transfer (Ramesh et al., 2022) to utilize a transformed target space given by an expert model (Figure 23). DeVise (Frome et al., 2013) used a pretrained skip-gram language embedding model and a pretrained deep convolutional neural network to embed images and their text labels as high-dimensional real-valued vectors in two distinct representation spaces. A translation model is then trained to map between an image-representation space to a label-representation space, thereby linking the spaces and allowing zero-shot se-

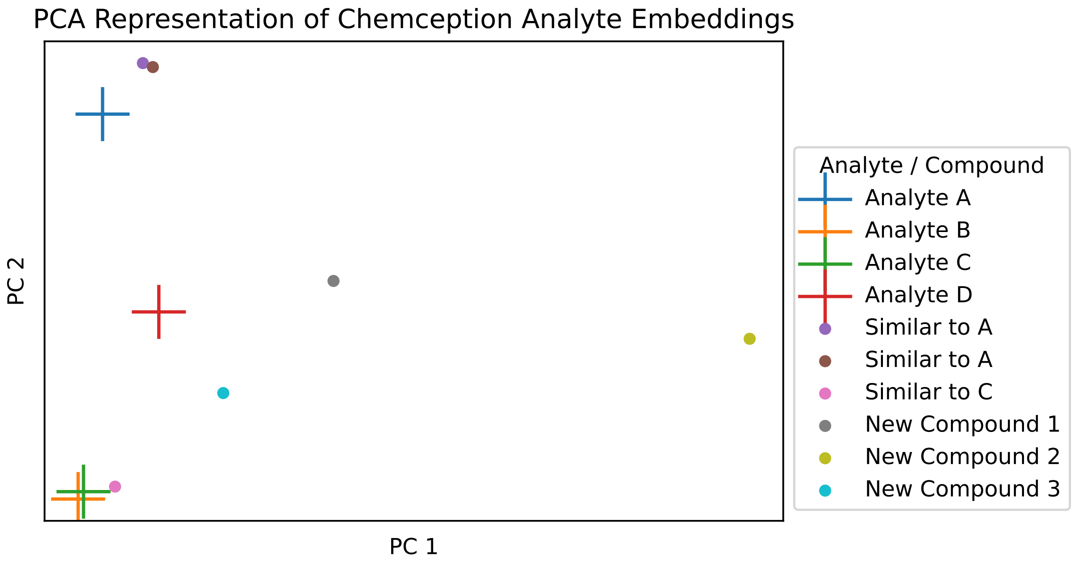


Figure 25: Embeddings of four analytes of interest as well as six comparison analytes and compounds. The penultimate activations of the Chemception model are used as embeddings and reduced to two dimensions with PCA for inspection. Chemception provides representations of analytes used as targets for the ChemVise embedder.

mantic prediction of input images.

There exist many pretrained deep representation models trained on large chemistry corpi mentioned in Section 1.3. In this work we utilize penultimate activations of the pretrained Chemception (Goh et al., 2017) supervised model which was trained to predict toxicity, activity, and solvation of a large dataset of chemicals. The inherent representation learning of deep neural networks provides representations of analytes we use to create a more meaningful target space for the ChemVise embedding model which trained from scratch for our novel chemical sensing hardware.

ChemVise uses targets represented by embeddings rather than traditional labels for a chemiresistive sensor signal input. Using representations of analytes as a target rather than typical one-hot classification or multiple regression targets leads to improved decision boundaries and signal representations and can be adapted to arbitrary sensing hardware. Figure 25 shows 2D visualizations of embeddings of analytes given by Chemception (Goh et al., 2017) encodings of analytes represented as images of SMILES strings match our intuition about similarity of analytes in a high-dimensional space. Representation spaces replace the label space as the target for a high-dimensional output[2].

A deep fully-connected neural network is trained to map from the input data space of sensor signals to this molecular-semantics representation space. Any supervised learning algorithm with a multidimensional real-valued output may be used. The ChemVise process provides a trained embedder which encodes any sensor response input in the information-dense space with improved separability between classes. Subsequently any classifier can be used in the chemistry space for improved classification outcomes (Figure 26) compared to non-embedded samples (Figure 18).

Each point of an analyte representation space encodes a similarity and dissimilarity to other molecule and compound points. ChemVise must provide a faithful embedding to the properties of chemical compounds as well as multi-analyte mixture signals. For this reason we implement linear combinations between single-analyte exposures with synthetic targets given by the geodesic center of the two canonical elements.

4.3 Linear Combinations Approximations

We utilize a natural image processing approach incorporating linear combinations to propose an application to multi-analyte signal processing. linear combinations

[2] Further detail on the implementation and training of ChemVise along with a PyTorch-style pseudocode is available in the extended materials section.

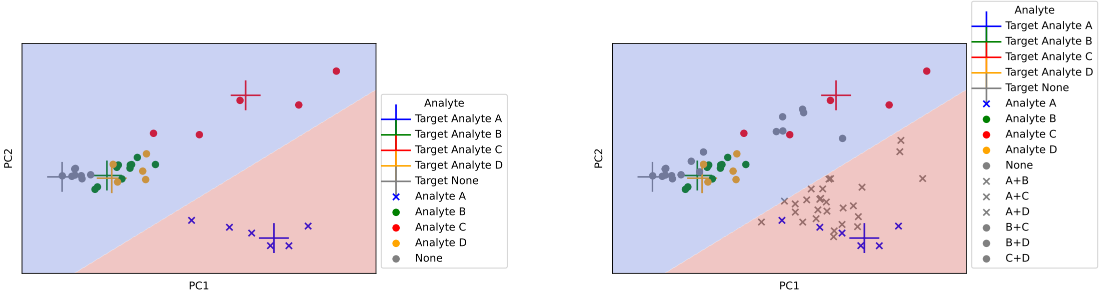


Figure 26: This visualization shows the validation singles and testing doubles (black) embedded by the ChemVise model then classified with a linear SVC. This image is for visual illustration only as ChemVise utilizes a 512-dimensional representation space, which is more accurate than the same model under a 2-dimensional PCA decomposition. **Left**: ChemVise embeddings given training the deep learning model to map signal samples to their corresponding target given by the matching color "+". The tunable decision boundary linearly separates the positive and negative samples. **Right**: Double-analyte samples under this trained embedding model fall in line with the correct classification given by the SVC trained with only single-analytes. This embedding outperforms simple decomposition embeddings by incorporating molecular semantics into the embedded representations, and allows even double-analyte samples to be linearly separated.

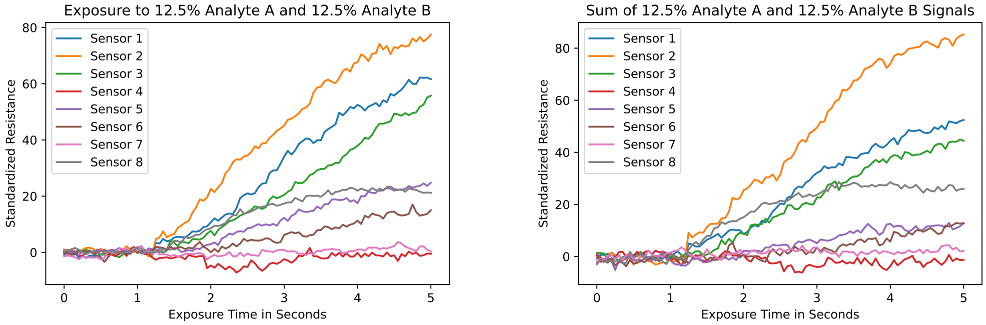


Figure 27: **Left**: Sensor experiment exposing eight chemiresistive sensors with unique polymer and graphene coatings to a mixture of 12.5% Analyte A and 12.5% Analyte B. **Right**: The summation of 12.5% Analyte A and Analyte B signals approximates the simultaneous double-analyte exposure to Analyte A and Analyte B. The synthesized double-analyte target in the analyte representation space is taken to be the weighted sum of the two single-analyte representation vectors according to the linear combinations parameter (Section 4.3)

is a natural image augmentation designed to promote simple linear behavior in-between training examples for deep neural networks (Zhang et al., 2017). The data-agnostic augmentation synthesizes samples given by linear interpolations between random examples and their training labels:

$$\vec{x}_k = \lambda \vec{x}_i + (1 - \lambda)\vec{x}_j \tag{14a}$$

$$\vec{y}_k = \lambda \vec{y}_i + (1 - \lambda)\vec{y}_j \tag{14b}$$

Where y_i, y_j are analyte exposure embedding vectors corresponding to sensor response vectors x_i, x_j.

Linear combinations as applied to chemiresistive sensing approximates real double-analyte chemical sensor exposures using only single-analyte exposures as training data. We set the mixture probability density function as a tunable uniform distribution between a and b. The maximum and minimum of the distribution may be tuned to account for what level of mixture sensitivity should be classified as a positive sample. Here, we utilize $(a, b) = (0.3, 0.7)$ as the bounds for mixtures representing the typical concentrations in the training data. Otherwise, utilization remains the same as in the natural image paradigm wherein samples are linearly interpolated during training using linear combinations of training pairs.

4.4 Results

In addition to single-analyte exposures of one analyte to the sensor array, multiple analytes may be exposed simultaneously. Here we refer to a two-analyte mixture exposed to the sensor device as a double-analyte exposure, and may be a combination of any two analytes with any relative magnitude. In our binary classification results 4.4, a mixture containing any amount of Analyte A is treated as a positive sample and all other mixture pairs are treated as negative samples.

We provide results for a training paradigm given by training on single-analyte exposures and testing on challenging out-of-distribution obscured double-analyte exposures. We demonstrate the importance of representation selection (Section 4.5) as well as optimized models for the rapid detection of obscured Analyte A given variable-length exposure windows (Section 4.6). Results within follow a consistent algorithm (K-fold validation hyperparameter gridsearch) for the selection of hyperparameters prior to observation of the holdout testing data.

4.5 Representation Matters

Though our design utilizes the pretrained Chemception (Goh et al., 2017) to represent analyte SMILES images as meaningful vectors in a target space, in theory any space could be used to represent analyte labels for embedder training. Here we compare the Chemception representation space with two alternative analyte representation spaces to use as targets for the embedding prior to SVC classification. "One-hot" denotes utilizing 512-dimensional (matching the Chemception embedding dimension) representations for analytes where signals containing Analyte i are represented as e_i, where e_i is the zero-vector with 1 at index i. "Equidistant spherical" denotes utilizing equidistant points from the 512-dimensional unit hypersphere as representations for the chemical analytes.

Figure 28 demonstrates the importance of the semantically meaningful space for the ChemVise embedding. Whereas one-hot and equidistant spherical representation spaces struggle to provide meaningful representation spaces for down-

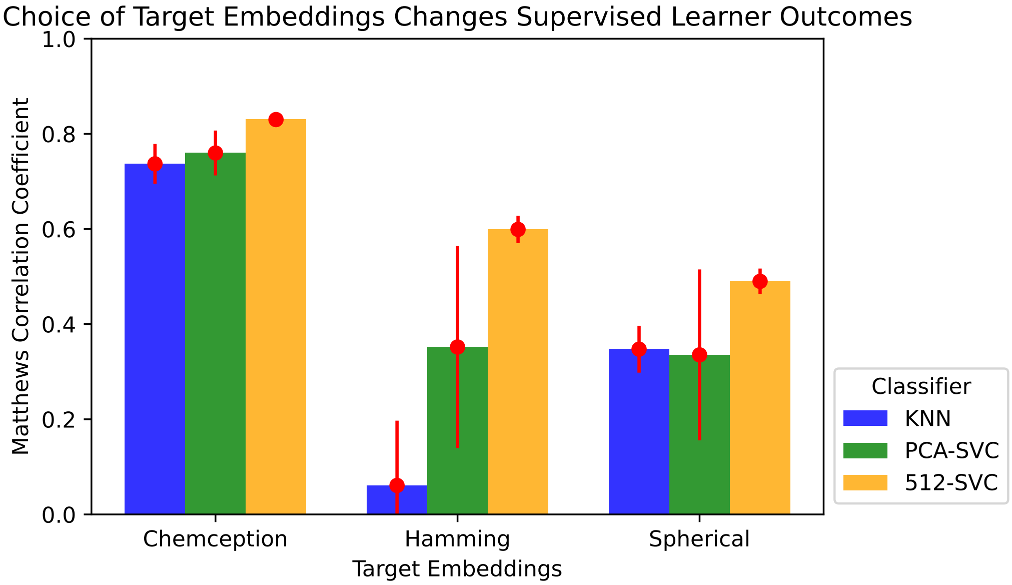


Figure 28: Classification metric on double-analyte exposures as a function of embedding model and downstream classifier. Downstream classifiers include a K-Neighbors classifier in the representation space, or a Support Vector Classifier in the representation or 2-dimensional projection of the representation space. The choice of Chemception, one-hot, or equidistant spherical targets modifies the training of the embedding deep learning model prior to classification by the classifier head.

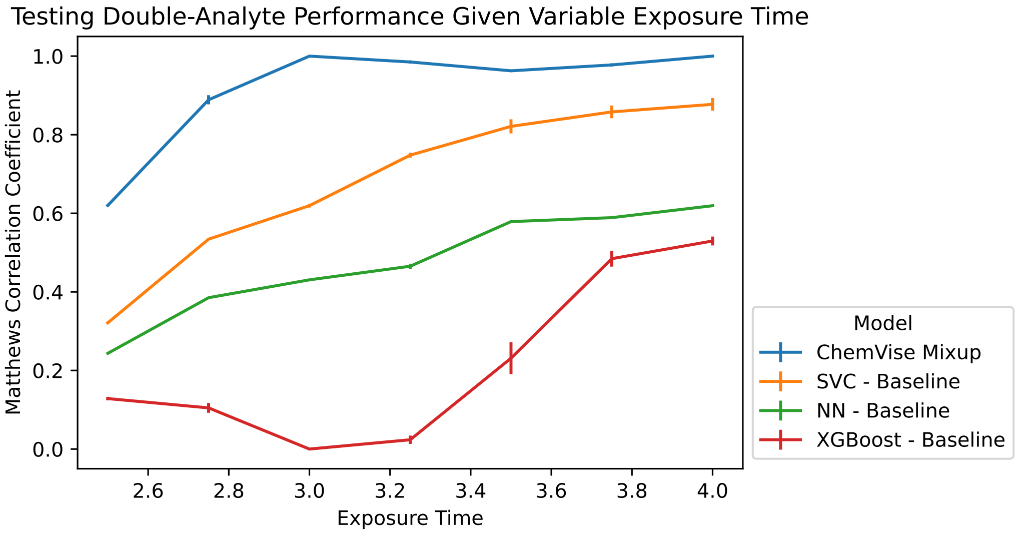


Figure 29: Matthews Correlation Coefficient score given the prediction of Analyte A in a holdout set of 39 real double-analyte experiments after training on only single-analyte exposures. Given five hyperparameter optimizations over five 5-fold cross validation experiments, model performances for 7 different exposure window lengths from 2.4 to 4.0 seconds.

stream SVC classification, Chemception consistently provides a high-quality representation space for three candidate classifiers. Semantic meaning implemented as intra-analyte distances dramatically improves the decision boundary outcomes over two approaches which do not incorporate external chemistry knowledge.

4.6 Variable Exposure Times

In Figure 29 we demonstrate results on the rapid classification of Analyte A in the presence of obscurant analytes with no corresponding obscured-analyte training data. Though multiple machine and deep learning approaches show a positive scaling behavior in the response metric as a function of exposure time, the ChemVise approach substantially and consistently outperforms the alternative models for the out-of-distribution detection task.

4.7 Conclusion

ChemVise representation learning outperformed linear decomposition embeddings as well as standard machine and deep learning supervised models by incorporating molecular semantics into the embedded representations. We demonstrated how improved chemical representations incorporating domain knowledge on molecular semantics for chemical sensing allow double-analyte samples to be encoded such that they are linearly separable with simple classifiers when only single-analyte real experiments are available. This approach may benefit researchers performing chemical detection with arbitrary hardware designs, and does not require pretrained models in the input domain. Rather, chemical classification may utilize sensor data from any sensors provided a corresponding target domain exists and is sufficiently studied.

Though transfer learning may be appropriate when the input data lies in a well-studied domain with learnable representations, the utilization of ChemVise-style target adaptation dramatically increases the number of tasks which may incorporate exterior data corpi. The ChemVise approach may extend the toolbox of a machine learning engineer performing supervised learning on a more diverse set of tasks with nonstandard inputs. The implementation would retain the deep embedding model and replace the target space with an appropriate pretrained model capable of representing the targets of the data set with some meaningful distance encoding. The generalizable approaches to supervised learning discussed here may dramatically improve sensor machine learning results and move away from complex, hand-tuned representations of domain knowledge.

5 The Great Chemical Sensing Bake Off

Multivariate time series classification (MTSC) problems are highly common in real-world applications (Ruiz et al., 2021; Gupta et al., 2020). The publishing of multivariate time series classifiers may outpace the rate of real progress in the improvement of classification performance across benchmarks. For this reason we investigate a broad array of approaches to multivariate chemiresistive time series classification including at least one representative from all popular algorithm types (Ruiz et al., 2021; Ottervanger et al., 2021).

5.1 Contest Rules

Each model is trained four times on each 75% split on the training data for each training data set. Then each of these split models is validated on a holdout set of testing data corresponding to that experiment. This is performed for each of 34 models (Table 30) and 11 chemical sensing data sets described in 1.3.

As in Ruiz et al. (2021), each of the bake off models utilizes the suggested parameters from the model authors. To account for discrepancies in training and validation time, we quantify the trade-off between training and performance in order to account for the benefits of hyperparameter tuning for the full bake off in Figure 37.

5.2 Bake Off Models

A diverse cast of models representing multiple approaches to univariate and multivariate time series classification are drawn for our chemical sensing bake off (Zhao et al., 2017; Dempster et al., 2020; Lubba et al., 2019; Löning et al., 2019; Middle-

Model	Type	Application	D
FullyConvolutionalNetwork	Conv. Deep Learning	Multivariate	Zhao et al. (2017)
ConvolutionalNeuralNetwork	Conv. Deep Learning	Multivariate	Zhao et al. (2017)
ROCKET	Random kernel transform	Multivariate	Dempster et al. (2020)
Catch22	Feature extraction transform	Multivariate	Lubba et al. (2019)
RandomInterval	Concatenates random intervals	Multivariate	Löning et al. (2019)
CanonicalIntervalForest	Catch22 on random intervals	Multivariate	Middlehurst et al. (2020)
HIVECOTEV2	Hierarchical transform ensemble	Multivariate	Middlehurst et al. (2021)
ShapeletTransform	Rotation Forest on shapelet transform	Multivariate	Bostrom and Bagnall (2017)
WEASELMUSE	Bag-of-patterns for intervals	Multivariate	Schäfer and Leser (2017)
BOSSEnsemble	Bag of Symbolic Fourier Symbols	Univariate	Schäfer (2015)
MatrixProfile	Unified motifs and shapelets	Univariate	Yeh et al. (2018)
ContinuousIntervalTree	Information based decision tree	Univariate	Deng et al. (2013)
RotationForest	Forest on random PCA transforms	Univariate	Rodriguez et al. (2006)
MultiLayerPerceptron	FCN Deep Learning	Non-temporal	Wang et al. (2017)
SVC	Support vector classifier	Non-temporal	Cortes and Vapnik (1995)
KNeighbors	K-neighbors supervised clustering	Non-temporal	Fix and Hodges (1989)
GaussianProcess	Gaussian process classification	Non-temporal	Seeger (2004)
DecisionTree	Information-splitting tree	Non-temporal	Quinlan (1986)
RandomForest	Forest of decision trees	Non-temporal	Ho (1995)
AdaBoost	Ensemble boosting errors	Non-temporal	Freund and Schapire (1997)
GaussianNB	Gaussian Naive Bayes classifier	Non-temporal	Chan et al. (1982)
QuadraticDiscriminantAnalysis	Quadratic classifier by Bayes' rule	Non-temporal	Rao (1948)

Figure 30: Competitive classifiers for chemiresistive sensor array classification, a brief description, and their origin.

hurst et al., 2020, 2021; Bostrom and Bagnall, 2017; Schäfer and Leser, 2017; Schäfer, 2015; Yeh et al., 2018; Deng et al., 2013; Rodriguez et al., 2006; Wang et al., 2017) (Cortes and Vapnik, 1995; Fix and Hodges, 1989; Seeger, 2004; Quinlan, 1986; Ho, 1995; Freund and Schapire, 1997; Chan et al., 1982; Rao, 1948). Each is given a brief description in Table 30. Univariate time series models are adapted to the multivariate time series classification paradigm with each of two algorithms detailed in Section 5.3. Non-temporal machine learning models are adapted using a tabularization of the multivariate time series data with column concatenation described in Section 5.3.

5.3 Bake Off Results

Figure 31 visualizes the mean performance of models across all eleven experimental data sets, with four splits of each training data set. Some models fail to converge for many data sets and have been removed from subsequent analyses (Matrix Profile Column Ensembles Ensembles, BOSS Ensemble Column Ensem-

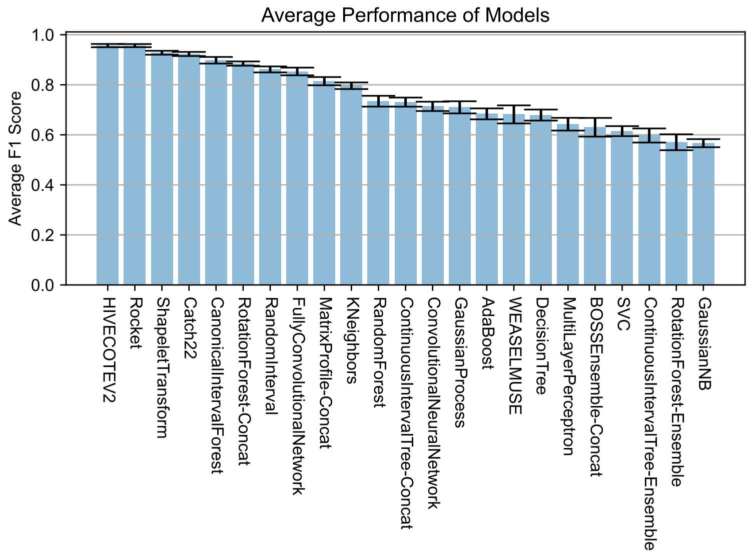


Figure 31: Average model F1 score and standard error on validation sets across 44 splits of 11 chemiresistive sensor array data sets.

bles, and Quadratic Discriminant Analysis). Our hypotheses on the chemiresistive data structure have assumed shapelets or time warping KNN would be the most successful models. Though these are effective, random transformations with linear classifiers (ROCKET) as well as complex ensembling (HIVECOTE) outperform shapelet-based approaches on average across models, despite failing to model the inductive bias from the training to testing set across a number of hardware arrays and data collection protocols.

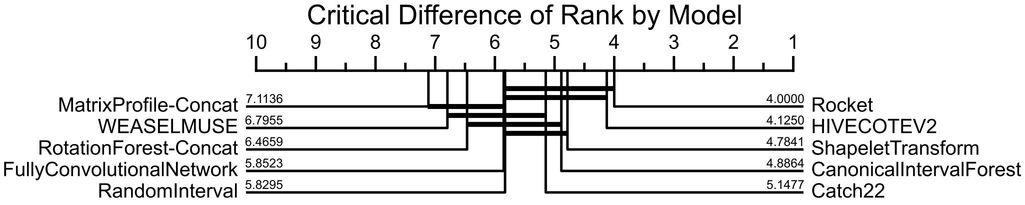


Figure 32: Critical difference plot demonstrating cliques of top 10 models by rank. Clique bars indicate statistical uncertainty of difference in average rank of clique members. Full critical difference available in Appendix Figure 45.

Figure 32 shows the critical difference between the top 10 model average ranks

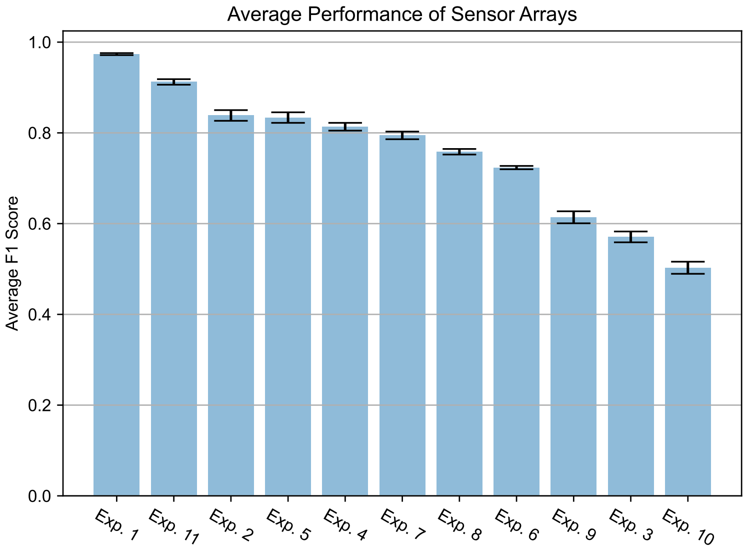


Figure 33: Average of all model performances (minus outliers) for 4 splits on each experiment. Differences in analyte discriminability due to varying sensor chemistries lead to a general score for the disciminability of chemical analytes under a certain sensor ensemble "lens." Formulating improved sensor chemistries increases analyte discriminability without regard for model inductive biases.

over 44 splits of the 11 sensor array training sets. The horizontal bands link a clique of models with statistically insignificant difference of average rank given performance across all splits. In the top-performing clique we find models from each of the primary families of time series classifiers - random kernels, feature ensembles, shapelets, decision trees, and deep learning (Gupta et al., 2020).

In addition to investigating the average performance of models across a variety of domain datasets, we are in a unique position in developing an effective chemical sensing tool at the hardware level. For our bake off, the optimization at the model selection step across datasets may lead to an effective model, but in addition we seek an effective dataset on which training leads to successful models in the discrimination of a target analyte. One approach to finding an optimal sensor array set is to compare the average performance of diverse models learning outcomes on the dataset (Figure 33). The corresponding testing accuracy of a model trained

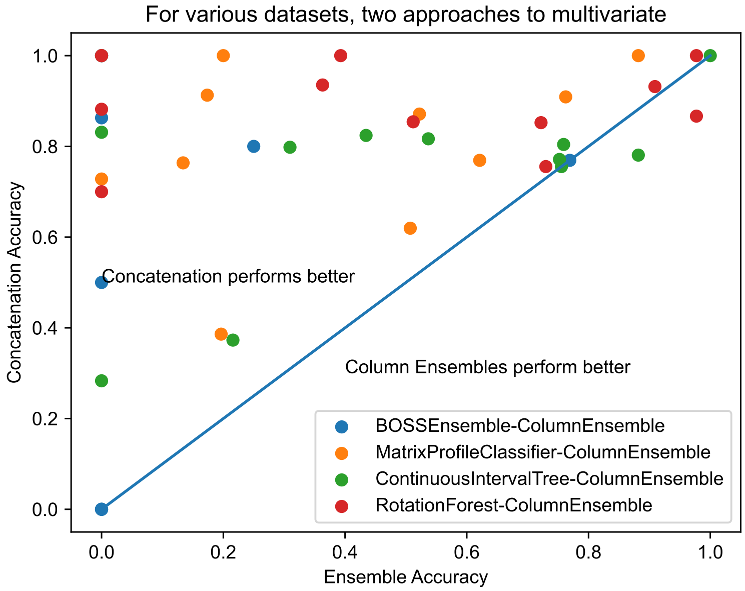


Figure 34: Two approaches to multivariate adaptation of univariate time series classifiers. Above the line, the column concatenation approach outperforms the column ensembling approach for the dataset and model. Hypotheses on the nature of sensor chemistries are confirmed by the necessity of multivariate information when determining class predictions, evidenced by the weak performance of column ensembling.

on a particular set gives a sense of the predictability and discriminability of the sensors, while ”factoring out” the inductive biases of each model by considering a highly diverse set of classifiers.

Univariate time series classification models are a well-researched literature and may be adapted to multivariate time series classification tasks with two alterations to the training algorithm of arbitrary univariate classifiers. These adaptations include the following:

1. Column Concatenation: given training data array $\mathbb{D}$ of n samples of k dimensions of length t (n, k, t) and classifier C, reshape the data by concatenating dimension d_i to the last element of dimension d_{i-1}, yielding array of shape $(n, k * t)$:

$$(n, k, t) \rightarrow (n_{1,1}, ..., n_{1,t}, n_{2,1}, ..., n_{2,t}, ...n_{k,1}, ..., n_{k,t})$$

Then train classifier C on resulting array $\mathbb{D}^*$ of shape $(n, k * t)$

2. Column Ensembling: given training data array D of n samples of k dimensions of length t (n, k, t) and classifier C, define k training subarrays $(n, d_1, t), (n, d_k, t)$ for each dimension d of k.

   Train classifier C_i on each resulting array $\mathbb{D}_i = (n, d_i, k)$, and use ensemble voting to classify the sample (Polikar, 2006).

These approaches have the advantage of relying on a rich literature of univariate time series classification, but each has a substantial downside. Column concatenation (1) substantially increase the dimension of the feature space, which is particularly disadvantageous for decomposition classifiers such as matrix profile classifiers as well as increasing the effect of the curse of dimensionality for high dimensional spaces. In addition this removes the option to early-classify samples (Chapter 6) and is incompatibility with state-based approaches such as an RNN. Second, the column ensembling approach (2) may train multiple classifiers which each fail to accurately predict the totality of the sample given one channel of information, as we assume with our chemical sensing data in which the *diversity* in sensor coating affinities yields discriminability. Individual sensors do not contain enough information for a classifier to make a reasonable prediction.

5.4 Rapid Classification

Research in the rapid detection of harmful chemical analytes often incentives rapid classification for improved safety devices. Here we propose a contest in the rapid

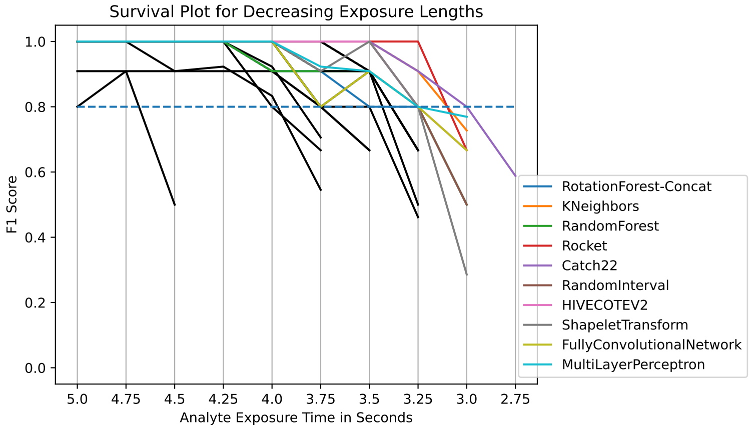


Figure 35: F1 scores of top-10 models trained on incrementally decreasing exposure times. Models which fall below an F1 score of 0.8 on the testing set for an exposure time are eliminated. As exposure lengths are incrementally decreased, models have reduced information via number of features to discriminate between analyte classes. Models which remain performant on short exposure times are highly feature-efficient in their decisions, and lead to rapid real-world detection when inference compute time is not considered.

classification of chemical Analyte A in increasingly challenging, foreshortened signals. Figure 35 visualizes the results of our experiment in the survival of models in retaining high classification accuracy during increasingly challenging rounds of chemical sensing by decreased exposure windows. At each iteration, the length of the exposure window is reduced by 0.25 seconds and all models are trained on the best set of experimental data, which we found to have higher average performance than the next best sensor array and data collection protocol. Early and rapid classification will be expanded thoroughly in Chapter 6 to include the literature on early classification of time series (Xing et al., 2012) and our novel architectures incorporating transfer learning representation spaces for semantic embeddings (Chapter 4).

An early classifier is *serial* if $\mathbf{C}[s[1, l_0]) = \mathbf{C}(s[1, l_0 + i])$ for any $i > 0$ - that is, the classifier **C** does not benefit from longer prefixes of the data and will not change the

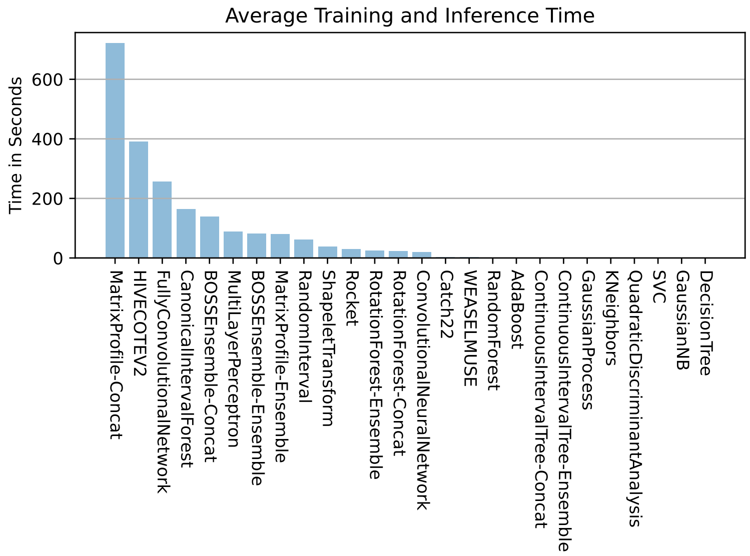


Figure 36: Wall-clock time in seconds to train and score each model on a training and testing set. Models with prohibitively high time to train lead to worse outcomes in the hyperparameter optimization step, where fewer configurations may be used. Models with prohibitively high time to infer fail to rapidly classify sensor signals.

estimate with further steps (Xing et al., 2012). Investigating Figure 35 in reverse, we find the best classifiers to be serial given exposure times of around 3.25 to 3.5 seconds, and do not benefit from further exposure to signals. Classifiers which remain serial to very brief exposure windows include ROCKET, Catch22, HIVECOTEV2, and RandomInverval classifiers. These represent a surprisingly diverse mix of shapelet-based approaches, model ensembles, and transformation classifiers, and demonstrates that shapelet-based approaches though inductively sound for the data may struggle with variance in sensor responses and generalizability.

5.5 Training and Inference Time

Figure 36 compares the time to train and test each model. Outlaying models include the MatrixProfile-Concat, HIVECOTEV2, and FullyConvolutionalNetwork.

The MatrixProfile-Concat training time demonstrates one shortcoming of the column concatenation technique of univariate classifier adaptation which is the significant increase in compute for dimensionality compared to the MatrixProfile-Ensemble. In addition the FullyConvolutionalNetwork would train significantly faster on a graphical processing unit, but specialty hardware may not be available for a real-world deployed tool with limited processing and power requirements.

Training time is highly significant for the hyperparameter optimization process as a model with 100x faster training may attempt 20x larger 5-fold cross validation hyperparameter sweeps than the more expensive model which could lead to a reordering of results under a wall-time restricted experiment.

An inference time plot also tells a very important story. When we deploy a hardware chemical sensing device, the amount of compute may be very limited. The time to process the inference of a sample would need to be factored into the rapid classification scores as the processing of the testing sample would delay reporting the classification result, resulting in wasted time in a safety situation.

To weigh the importance of time to train and test against accurate predictions, we visualize the spread of model training time and testing performance in Figure 37. We see various outliers in the time to train with significant competition from models with slightly lower accuracy but orders of magnitude faster time to train. ROCKET and Catch-22 are highly promising in this time-corrected performance.

5.6 Discussion

A bake off is a fundamental cornerstone of applied machine learning. Addressing the inductive biases of a variety of models is particularly motivating when we also have access to a variety of experimental protocols and sensor coatings, which

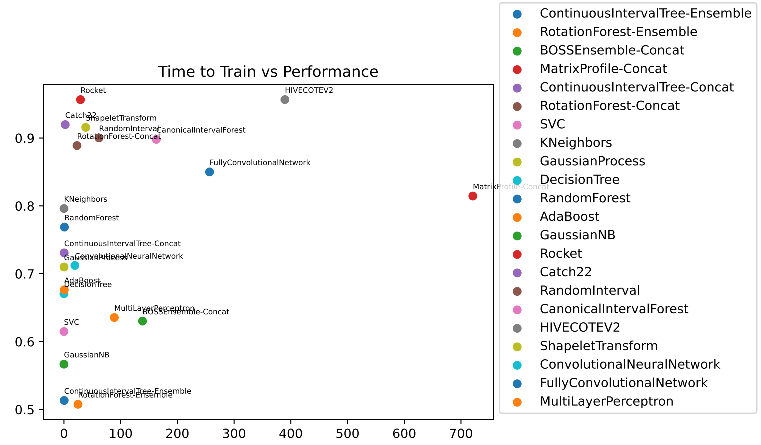


Figure 37: Time to train vs. average model performance across sensor arrays. Models with higher compute costs at training time tend to outperform those with lower training time on holdout data. This efficiency curve has optimal points with a few modern, efficient classifiers.

themselves make assumptions about the nature of chemical discrimination. By replicating the experimental design of Ruiz (Ruiz et al., 2021), we find that as a trend models which perform well on the diverse UEA archive (Bagnall et al., 2018) of multivariate time series classification data sets also perform well on our chemical sensing task.

Additional nuance in bespoke chemical sensing models gives a highly promising direction for future research. Chapter 4 may by adapted from the multiple classification task on input signals to a multivariate time series classification task leveraging semantic information in a chemistry-informed representation space to form meaningful clusters through time. These sequences of unsupervised clusters will also call upon the outlier detection and out-of-distribution literature. By first training a semantic representation model, the meaningful embeddings may yield greater inference for unsupervised learning, clustering, and out-of-distribution analyses.

6 *ChemTime*: Early Classification of Time Series

This chapter will make use of animations in order to demonstrate the real-time classifications, representations, and decision boundaries for multivariate time series classifiers on a real-world chemical sensing task. Animations corresponding to figures of this section may be found online at `https://github.com/alexander-moore`.

6.1 Introduction

Model benchmarking in Chapter 5 does not tell the complete story about model performances on hardware sensor arrays. Generalization to more complex exposure types (Chapter 3), time to train and inferential time, and explainability are also commanding forces of model selection for a real-world chemical sensing tool. Though previous chapters have explored the topic of rapid classification for low sequence length exposure times, *early* classification learns adaptable versions of single models to classify unlabelled samples at a confident stopping time. In this chapter we propose a novel extension of the *ChemVise* paradigm for the early classification of multivariate chemiresistive sensor time series, and demonstrate how using known biases of the targets and data dramatically improve the performance for this unique classification task.

Some classifiers introduced in Chapter 5 have limitations which preclude them from deployment to a real-world hardware sensing tool. Algorithms requiring fixed-length time series cannot be adapted to variable-length sequences without sequence padding, and cannot be extended to early classification. Some approaches are non-parametric and computationally expensive in processing or memory requirements, particularly for limited throughput onboard computers in a deployed tool with power constraints. For a detection paradigm where one second of chemical exposure can have dramatic consequences, even highly accurate models may be inappropriate if they come with long detection times. In addition, domain experts often require additional levels of confidence, explainability, and inference which some black-box models such as the deep learning suite and some machine learning approaches fail to offer.

As a trend, many time series classifiers bootstrap the input data to generate multiple transforms, sets of summary statistics and feature analysis. However, these approaches require processing an entire signal of constant input length. Often in time series applications a classification estimate is required in a shorter time frame, or the length of the input sequence is variable. In addition there may be an informative signal relevant to the class label early in the signal which is followed by irrelevant information, potentially for a long time. In this situations it is vital to turn to early classification of time series (ECTS). Early classification corresponds to returning the predicted label of the sample before the final feature of the signal is observed. These models require some decision function which determines if the classifier has observed enough information to return the predicted class early. Additional results in this chapter will investigate rapid classification: observing complete signals which have been intentionally foreshortened as an alternative to the early classification paradigm.

Previous results in Chapters 3, 4, and 5 have noted the importance of *rapid* classification on multivariate time series data from chemiresistive sensor arrays. Each of these chapters includes results for the performance of models as we manually reduce the length of the exposure window, corresponding to more rapid detection of the chemical analyte in wall-clock time as well as classification on a reduced number of features. Chapter 6 will utilize the *early* classification literature and contrast the ideas of rapid and early classification for this task with diverse algorithms for models which self-select their stopping times with a variety of confidence measures and differentiable mechanisms.

In addition to the early and accurate classification of particular chemical analytes under a chemiresistive sensor array, this chapter will outline the explainability and inferential advantages and disadvantages to a number of models to arrive at a highly interpretable embedding model with multiple utilities to ongoing chemical detection research.

6.2 Early Classification of Time Series

In a seminal paper on the early classification of time series, Xing et al. (2012) proposes a simple approach to the early classification of time series using a one-nearest-neighbor approach (1NN). At training time each observation is labelled with a minimum prediction length (MPL) corresponding to the timestep at which the sample may be assigned to one class. At inference time, time series s is classified by ascending through time series values i and returning the dominating class label of time series of 1NN(s) that have an MPL of at most i. To generalize, they learn a reasonable stopping point for the time series given by the point at which samples may be classified as determined on a validation set.

For the application of early classification to chemiresistive sensor arrays and obscured chemical analytes, this mechanism fails in a few ways. The training data may limit the earliness of detection as inferential samples must wait until the MPL of the training data (Xing et al., 2012). Second, the 1NN approach may overfit a training set for high-noise sensors or obscured analytes. Earlier results in the detection of obscured chemical analytes using our sensor arrays demonstrate that KNN-based approaches including the improved DTW algorithm fail to learn generalizable representations for the holdout single and double-analyte exposures (Chapter 3, Bagnall et al. (2017)).

Xing et al. (2012) address these 1NN shortcomings in their Early Classification on Time Series method (ECTS). Their approach to ECTS forms clusters on training data in each prefix space resulting in a sequence of clusters through time. Stable clusters may be formed at training time, then report when a new point joins a known stable cluster at inference time. This method however requires taking pairwise distances between the inference sample and the entire training set at testing time. This is a non-parametric approach which makes no inductive biases on the data, but may require extensive computation at inference time to determine cluster membership at each time point.

One shortcoming of the ECTS approach (Xing et al., 2012) is calculating distances between signals in the time-series space. This means we can expect classifiers to perform similarly to KNN classifiers over time series, such as Dynamic Time Warping (DTW, Bagnall et al. (2017)). DTW significantly outperforms KNN due to accounting for varying distance metrics and is a high quality benchmark across UEA MTSC archive (Bagnall et al., 2018). However, Bagnall et al. (2017) finds that DTW today is only a baseline, and modern approaches can substantially outperform even time-warping based KNN. Likewise, we anticipate modern multivariate time series classifiers in conjunction with representation learning for ECTS may substantially improve early classification outcomes.

6.3 Contemporary Early Classification Literature

Other research has investigated parameterized approaches to early classification on time series using differentiable neural network mechanisms. Hartvigsen et al. (2019) utilizes a bespoke LSTM network architecture with a learned early decision criterion. Rußwurm et al. (2019) proposes a similar end-to-end-differentable parameterized approach using distributions of stopping probabilities over the time series with an additional output probability density function head applicable to any architecture. Other approaches to parametric multivariate early classification are included here with a brief discussion on their relevance to our methods and chemical sensing task.

Hsu et al. (2019) proposes applying the modern attention mechanism for deep learning ECTS. Shortcomings with attention may include known long sequence lengths before positive class samples are seen, and attention between discrete regions of the multivariate signal may not be appropriate for our sensor detection task.

RelClass Parrish et al. (2013) is a probabilistic framework using quadratic discriminants and support vector machines for ECTS returning the degree of confidence with which one can say that the current incomplete data is sufficient to come to the same classification as the complete data with high probability. Likewise, Mori et al. (2017) employs a similar method which includes a metric for prediction reliability. These techniques are comparable to Xing et al. (2012)'s ECTS in the earmarking of signal subsets at training time for lookup at inference time. We may utilize this probabilistic framework on signals projected into our parameterized latent space sequences from Chapter 4.

He et al. (2020) supplies a confidence-based model for multivariate early time series classification with multiple interpretable rules. This approach emphasizes explainability and confidence of estimates but does not make use of parameterized learning for the classifier, and does not have a representation layer which facilitates transfer learning. Adapting the methods from confidence-based models in ChemTime's latent may be a fruitful direction.

Ottervanger et al. (2021) provides a small survey of nine multivariate early time series classifiers (some are included here, including Hartvigsen et al. (2019), He et al. (2020) and Xing et al. (2012)) and includes implementations which we may draw from for a bake-off style to compare our *ChemTime* approach to in future results sections. Ottervanger et al. (2021) makes the additional step of an automated machine learning approach to maximize earliness and accuracy for each model in an automated grid-searching pipeline.

Ismail Fawaz et al. (2019) surveys time series deep learning without early classification. These bespoke time series neural network architectures and training paradigms may be relevant for chemical sensors and adaptation for neural network early classification as we saw in 5. Deep learning multivariate time series classifiers may be adapted with Rußwurm et al. (2019) to append the end-to-end differentiable early stopping probability density function, or with *ChemVise* (Chapter 4) transformations in conjunction with latent early classifiers.

6.4 *ChemTime*

We propose *ChemTime* as an adaptation of the transfer learning techniques from Chapter 4.2 by incorporating contemporary research on early classification to a semantic-prediction time series representation model. We hypothesize that the improved representations measured by improved supervised learning outcomes

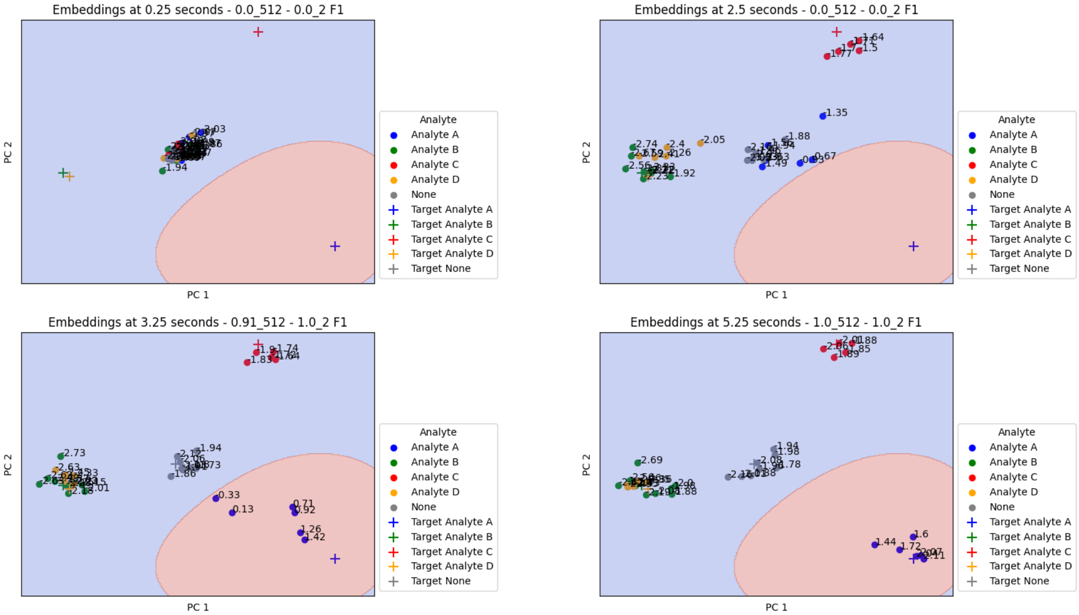


Figure 38: PCA Visualization of *ChemTime* latent representation through time for a dataset of chemical sensor data. As the RNN component of *ChemTime* iterates over a chemiresistive sensor signal, the state vector is projected into Chemception's activation space. The sequences of projections yield an inferable, predicable representation for the boosting classifier. Full animated figure available at `https://github.com/alexander-moore`.

for out-of-distribution double-analyte data will remain true for time series and early classification data in future experimentation. We additionally incorporate elements of the literature on early classification Xing et al. (2012); Hartvigsen et al. (2019); Rußwurm et al. (2019) with time series neural network architectures (Ismail Fawaz et al., 2019; Ruiz et al., 2021) to make early classifications in the chemistry-informed meaningful latent.

Our approach benefits from inferential machine learning techniques due to meaningful semantic embeddings rather than an activation space or simple class predictions. Experts can manually inspect linear projections of the representation space where regions and distances are informed by molecular chemistry. We can induce measurements on the space such as confidence given by distances to deci-

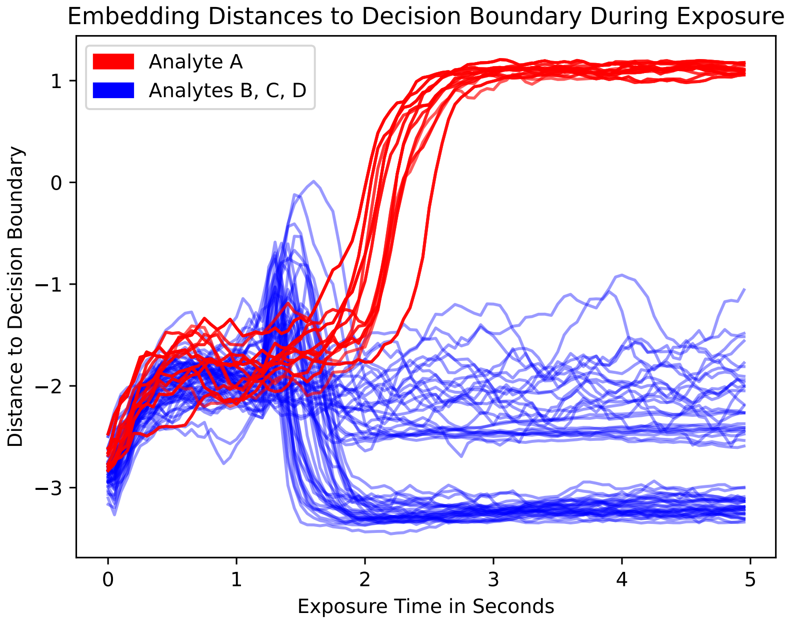


Figure 39: Distances through time from boosting classifier decision boundaries give inference over samples. A validation set may determine the optimal early classification window for testing set samples based on embedding trajectory.

sion boundaries and distance to training samples. SVCs fit on untransformed data could yield this metric but are not performant on untransformed data (Figure 18). Wiens et al. (2012) has previously used distance to the SVM as confidence in the classification, but using distances to samples for a confidence measure in *representation space* is yet unexplored to our knowledge.

Figure 39 visualizes this advantage by plotting the distances of the embedded samples from a boosting classifier's decision boundary. As time passes in the exposure signal, positive class samples tend to move in the positive class direction as negative class samples move into the negative direction. In this manner complex multivariate time series signals may be summarized into a single sequence given by distance. Simple 1NN classification for ECTS is sufficient to induce early classification on this model Xing et al. (2012). Using techniques from Section 6.3, an early-decision threshold could be determined using a validation set. When samples in a new testing distribution pass the decision threshold, the classification is returned regardless for the current timestep. In this manner, techniques from the established **univariate** early classification literature (Xing et al., 2012) can be used in conjunction with the *ChemTime* projection for representation sequences.

Representation targets given by Chemception embeddings (Goh et al., 2017) serve as cluster centroids known prior to training informed by molecular representation. Distances to these centroids, and mean pairwise distance from other clusters in the embedded training set give us a sense for the discriminability of analytes at each time point in the resistance curve. Xing et al. (2012) emphasized the minimum prediction length (here, exposure time) of samples in the training set such that the label could be uniquely determined by the 1-nearest neighbor classifier. Here, we may determine the sequence length such that the testing sample confidently belongs to one analyte cluster given only simple linear projections on an internal state to a representation layer.

A recurrent neural network is an iterative model which embeds each step of the input series according to the new set of features and the current internal memory. This begets a simple representation update at each time step of the input sequence. This sequence of representations also yields a class prediction, estimate confidence, cluster membership, and visualizable representation for each time at inference, each for a single linear operation given by a projection of the concatenation of the state vector and new time step features. This begets a further advantages of *ChemTime*: the possibility for open-ended real time classification of time series signals. For arbitrary signal lengths, at any sampling frequency with two linear operations the internal state is updated and the new representation is classified with a simple boosting classifier, yielding extremely low-latency responses investigated in Section 7.1.

Xing et al. (2012) fits a 1-Nearest Neighbor classifier given by early prediction time on the training set at each time step, including at inference time. Taking pairwise distances between suffixes of the entire training corpus may be intractable for long sequences or limited compute devices. For our detection device and lightweight edge device computation, this calculation may be intractable. Instead, parametric representations learned from the data as in *ChemTime* may encode knowledge from an arbitrarily large training corpus for arbitrarily large sequence lengths without increasing computation expenses. Bespoke neural network architectures for time series processing alleviate these shortfalls in previous approaches to time series classification.

6.5 *ChemTime* Implementation and Design Considerations

Training a deep learning representation of time series data may be done in a variety of supervised and unsupervised ways. Inducing novel loss functions motivated in Chapters 2, 3, and 4 give us multitask classifiers for unsupervised representations of time series, as well as semantic training for improved supervised learning through transfer of chemistry knowledge to our chemiresistive sensor task.

Multiple novel architectures facilitate the application of time series and transfer learning paradigms to our chemical sensing task. One important departure from earlier chapters is the inclusion of recurrent layers which corresponding to the inductive bias of time series data (Section 1.5). Recurrent layers iteratively process features through the time dimension of a sample and update a learned memory state at each time. In this manner the network has a limited bandwidth memory for ongoing representations of the sample through the signal. We may utilize these representations through time as embeddings of input samples for each time step.

```
# Require:
# chemception_enc: Chemception with final linear layer removed
# RNN: One-layer iterative neural network
# lm: Linear projection from RNN state to target representation dimension
# chem_start: time index where analyte exposure begins
# criterion: Euclidian distance, cosine distance, DEViSE distance

for signal, label in train_loader: # load a batch of labelled sensor samples
    # Encode training labels to analyte-space. Before analyte flux begins, label should be an innate gas
    y_rep_seq = [chemception_enc(label) if i > chem_start else chemception_enc('Nitrogen') for i in signal.sequence_length]

    # Iterate over sequence with RNN
    x_seq = RNN(x)

    # Project RNN state sequence to representation space
    y_hat_seq = [lm(x) for x in x_seq]

    # Classification metric to target embeddings
    loss = criterion(y_hat_seq, y_rep_seq)

# Downstream classifier on embedded signals
# Use final output of projection for boosted input
emb_xtrain = lm(RNN((x_train))[-1]
opt_model = SVC.train(emb_xtrain, ytrain)

```

```
# Use the classifier to predict final representation of testing signals
emb_xtest = lm(RNN(x_test))[-1]
emb_ytest_hats = opt_model.predict(emb_xtest)
test_score = classification_metric(emb_ytest_hats, ytest)
```

Pseudocode 6.5 demonstrates a simple training loop for a PyTorch-style implementation of *ChemTime*. *ChemTime* works with an inner training loop which predicts the label of the sample at every time step of the signal. Unlike normal RNN training, the loss is taken at every time step and aggregated over the time steps of the batch of signals. This yields a model that at each time step updates the internal state representation. This internal state is projected to the chemistry-informed latent with a simple linear projection learned during the optimization. Labels are manually crafted on the laboratory training data such that ”no analyte” is the label until the known analyte is flux begins (Section 1.3).

ChemVise Moore et al. (2023) introduced a molecular-semantic latent space styled on combined natural image-natural language spaces as in DeViSE Frome et al. (2013) and DALL-E Ramesh et al. (2021) for improved classification of out-of-distribution chemical analytes. Though *ChemVise* demonstrated how a chemistry-informed latent space using a pretrained molecular target embedding model improved classification outcomes over baselines, it failed to leverage inherent time series structure in the data.

ChemTime modifies the *ChemVise* approach by replacing the tabular-data embedding model with an iterative time series embedder with a moving-target approach to signal embedding. At each time step, the iterative encoder model uses a recurrent neural network backbone to encode the resistance signals of the input. At each iteration, a linear projection layer maps the state vector to a point in the molecular embedding space. The model loss is given by the sum of losses over the sequence of timesteps t, $\mathbb{L} = \sum_t \mathbb{L}_t$ where $\mathbb{L}_t$ may be given by an embedding distance such as a cosine distance, euclidean distance (summation of mean squared error losses), or bespoke hinged representation loss as in DeViSE Frome et al. (2013).

ChemTime has multiple benefits over the predecessor architecture. Utilizing sequences of representations in a meaningful latent space yields improved classification outcomes, inference during testing, and earlier classification compared to the fixed-window approach. Results sections will demonstrate the benefits of a light-weight approach for rapid classification, and the benefits of inference and analysis when sequences of representations may be used for meta-classification in the chemistry-informed latent.

Label Sequence Generation

ChemTime requires rephrasing a chemical exposure label into a sequence of targets in a chemistry representation space. Data are provided with labels corresponding to the concentration of the analyte exposure. For example, an exposure of 17% Analyte B will be labelled [0,17,0,0]. We build a sequence of targets by first getting the representation of Analyte B as the activations of Analyte B in a chemistry representation model. For the purposes of *ChemTime*, we utilize *Chemception*, though any molecular representation could be used Goh et al. (2017). This representation is then unrolled into a sequence by concatenating a 'None' representation (the activations of the inert gas Nitrogen under the same molecular representaion model) for each time step in which the flux of analyte vapor has not begun, and the representation at all subsequent time steps.

RNN and Linear Projection

ChemTime uses a two-stage projection to iterate over a resistance signal and map to the chemistry-informed latent space. The first is a RNN which concatenates the feature vector at time index t with the state vector from the previous time step. This is passed through a simple linear layer to return the updated internal state for the next timestep.

Boosting

Subsequent to training the *ChemTime* sequence embedding model, a tabular machine learning model may be fit on the final representations in the sequence. This is to classify the samples, which can be done with a naive nearest-target approach, or with a learned decision boundary. *ChemTime* results discussed here use a simple SVC with a binary decision discriminating Analyte A samples from samples containing Analytes B, C, and D.

7 Results Against the Field

In order to provide demonstrate the performance of *ChemTime* against a relevant field of effective multivariate time series classifiers, we evaluate the model performances across a diverse set of real-world chemical sensor hardware experiments. We estimate classifier performance for a general chemiresistive sensor array by studying eleven different sensor configurations, each with unique surface chemistries as in Chapter 5. Model performance on these eleven distinct data sets demonstrates the efficacy of each classifier family for the broader domain of chemiresistive sensor array classification.

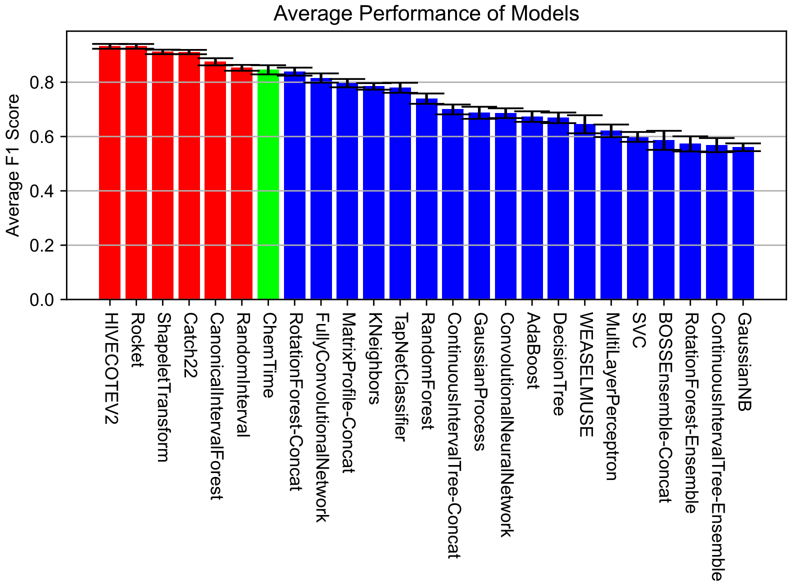


Figure 40: Average model performance measured by F1 testing-set score across 44 splits of 11 chemiresistive sensor array data sets. Each model from Chapter 5 is benchmarked and color-coded with respect to the average performance of *ChemTime* across the testing distribution.

We study the supervised performance of classifier families with the following process developed in Section 5.3. Each classifier is trained four times on each of eleven real-world chemiresistive sensor array data sets. Each of these four training iterations uses 75% split on the training data where a 25% fold is removed from the training corpus for that classifier split. Then each of these split models is validated on a holdout set of testing data corresponding to that experiment which does not include training split samples nor withheld split samples. This is performed for each model described in Table 30, *ChemTime* and 11 chemical sensing data sets described in Section 1.3.

Figure 31 visualizes the mean performance of models across all eleven experimental data sets, with four splits of each training data set. We situate *ChemTime* as the green bar against which all other models are contrasted. In terms of direct average performance across 44 splits, we find *ChemTime* to be the absolute 7th ranked

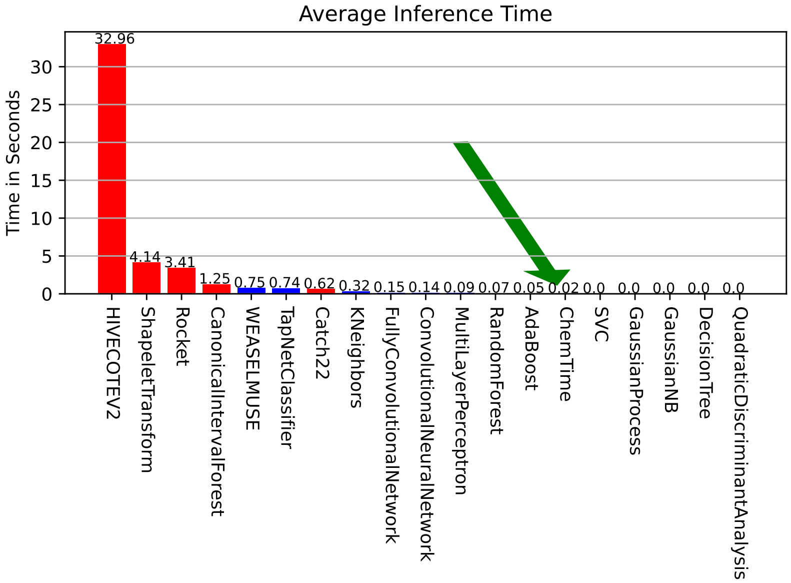


Figure 41: Wall-clock time in seconds to test each model on a testing set of 32 hold-out trials. The height of each bar represents the time each model takes to process a batch of testing trials and return a classification on the presence of Analyte A. Extreme outliers exist in the processing time of a 5-second exposure signal. We accentuate the low inference time of *ChemTime* with a green arrow, and notice that high-performing red models are correspondingly high in inference time.

model against the competition of general classifiers.

Figure 41 visualizes the wall-clock time needed to compute the prediction of a five-second exposure for each model in the study. We find that some models inference time excludes them from application to this task. Models which outperform *ChemTime* plotted in red all take substantially longer to perform a prediction. HIVECOTE stands as an extreme outlier which takes over six times the length of the signal in seconds to process the batch. As part of a deployed tool, we expect rapid classification. This requirement rules out a large number of models despite their high performance.

This chapter introduced the relevance of classification computation time to this chemical sensing task. In order to find models which optimally balance rapid computation and accuracy across a field of chemical sensing data sets. These findings

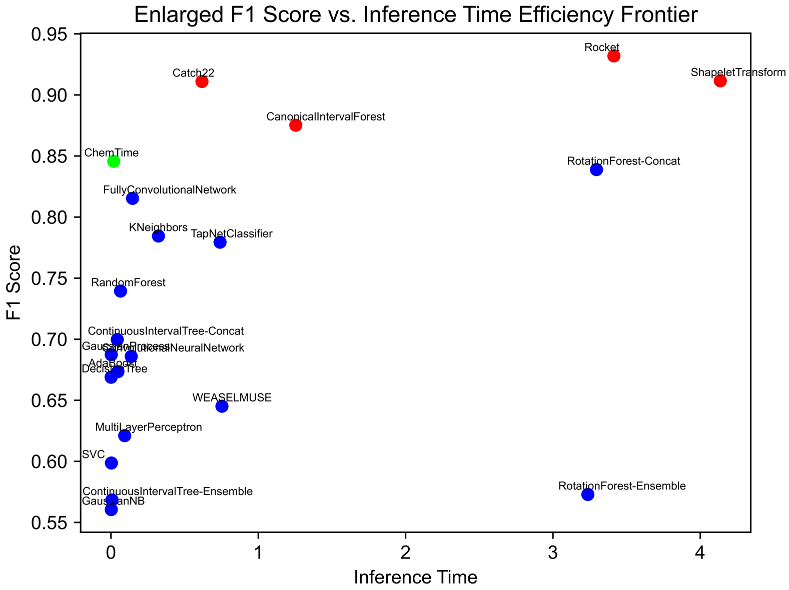


Figure 42: Time to infer vs. average model performance. An efficiency frontier exists between the inference time in seconds for sample labelling and the average model performance on a testing set. *ChemTime* appears to improve on the efficiency frontier by responding faster than the existing frontier.

are visualized in Figure 42, with outliers in computation time and performance removed. We find there exists an efficiency frontier in the trade-off between computation time and performance: two factors which are highly relevant for the detection of chemical analytes. Though some models are inferior in one metric with respect to another nearby model, there does exist a trade off between computation time that must be sacrificed to improve accuracy, and vice-versa. We note that *ChemTime* appears to improve this frontier from the baseline set in Chapter 5.4.

7.1 Rapid Classification

Rapid classification is vital for a successful deployable chemical sensor. Two factors contribute to balancing a fast and accurate model predicting incoming resistance data. First, the length of the analyte exposure. Longer analyte exposures

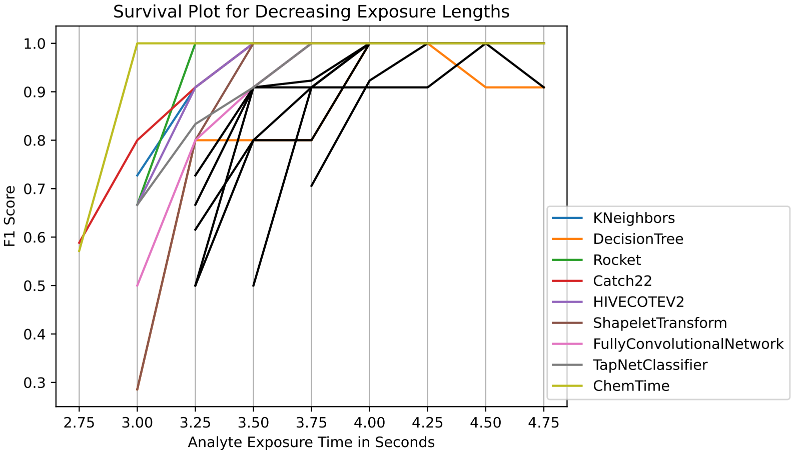


Figure 43: F1 scores of top-10 models trained on incrementally decreasing exposure times. Models which fall below an F1 score of 0.8 on the testing set for an exposure time are eliminated.

contribute to increased classification performances until models become *serial*, or do not improve given more information about the signal. The point at which classifiers become serial is vital to discovering data-efficient classifiers capable of rapid classification. In a deployed chemical sensing tool, the ability to respond quickly given a short analyte exposure is vital. This period of response is measured in two ways: the inference time (Figure 41) which measures the period of computation from observing the signal to reporting a classification, as well as the signal exposure time. Both of these time periods must be considered when selecting a rapid-response model.

In order to study which models consistently perform well despite a challenging paradigm of short sensor-exposure times, we propose the ”survival plot” Figure 43. In this plot, models are trained on 4.75-second exposures from the chemical sesning training data. Each model is then scored on a holdout validation set. Models which maintain a F1 performance metric above 0.8 on this validation set remain in contention, while underperforming models are eliminated. This iterative process

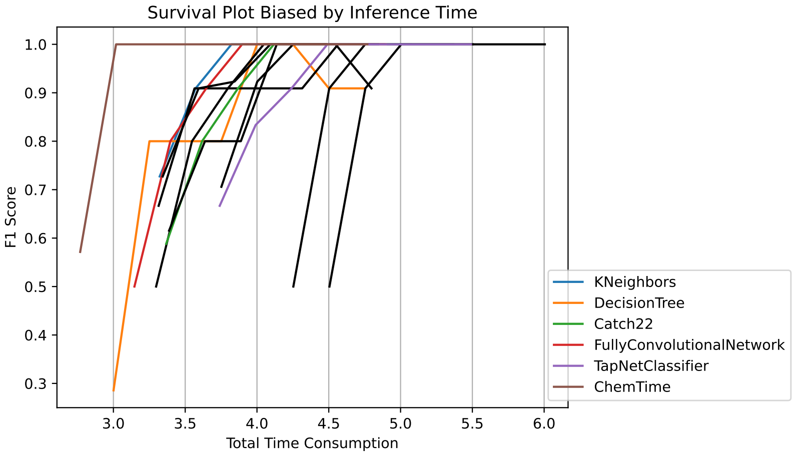


Figure 44: Survival plot biased by the inference times factored into the length of exposure. When accounting for the processing time of the model as part of the length of exposure, the improvement of *ChemTime* is remarkable. Furthermore, *ChemTime* benefits from real-time signal processing which many others do not.

is summarized in Figure 43 in which model performances drop off as a function of the reduced exposure windows, where all models fail to remain accurate given 2.75-second exposures. Many models remain serial from 3.5-seconds onward.

After observing some period of analyte exposure signal, some amount of computation is required to return the predicted label of the sample. The time this computation takes is a vital element to the rapid classification of chemical analytes in the real world, where short time delays in positive class identification could have devastating consequences. Therefore there are three key elements to identifying successful real-world models for a deployed chemical sensing tool: the amount of time the signal must be observed, the amount of processing time to return a classification, and the accuracy of that model in predicting the correct label within that time frame. Figure 44 summarizes this trivariate optimization by modifying Figure 43. Each curve from Figure 43 has been biased by the computation time the model requires in the total exposure time direction. The utility of *ChemTime* in this paradigm is even more apparent: when correcting for the total inference time of each model, the extremely rapid classification of *ChemTime* makes the success of the model stand out from other accurate models which require substantially more computation time. In the real world, this faster computation means *ChemTime* could observe more of the signal (and harvest more information) to make more accurate classifications while responding within the same time constraints as other models. These benefits would be even greater if *ChemTime* was implemented into real-world hardware such that the time series iteration was performed in real time.

8 Future Inclusions

ChemTime represents a significant improvement against the field of general-purpose multivariate time series classifiers for the domain of multivariate chemiresistive sensors for chemical sensing. By modelling biases from the data into the architecure and incorporating domain knowledge in the target space in order to induce a boosting classifier we substantially improve the time efficiency, performance, and rapid classification of unseen analyte samples. However, these findings only invite further research on the modification of *ChemTime* architecture to include improved representation, early classification techniques, and outlier detection for sequences.

ChemTime, inheriting from *ChemVise*, draws upon the conjoined image-language representation space of DeViSE (Frome et al., 2013). This joined representation space for the chemical sensing application presently uses the Chemception (Goh et al., 2017) activation space. A drop-in improvement for *ChemTime* representations of molecules in the codomain may be an improvement. Furthermore, DeViSE and the later more popular CLIP (Radford et al., 2021) latent advertise learning a conjoined representation space for the improvement of out-of-distribution classification. Substantial ongoing study could be dedicated to the classification of chemical analytes which do not belong to the training distribution. Due to the relationship between the molecular-semantic latent and the behavior of chemiresistive sensors, we may expect interesting findings in the study of zero-shot or few-shot learning using the *ChemTime* paradigm. Further research may investigate data synthesis leveraging similar hypotheses on the nature of these latents.

Though iterative representations of samples are highly motivating for unsupervised clustering in latent spaces, learning meaningful representations requires a meaningful loss over the training samples. There are many meaningful losses for the network to learn time series representations of input samples, particularly when drawing from Chapters 2 and 4. These include cycle reconstructive losses, adversarial losses, and semantic training losses. These losses may be further extended to address the representation loss. Though presently a summed MSE loss over the sequence, an improved loss function modelling the importance of a rapid classification may improve performance.

Finally, the choice of boosting model begets a great deal of exploration and experimentation for the future of *ChemTime*. Boosting models could be brought in depending on the task at hand, and could be an ensembling of multiple models for maximum performance on a variety of tasks. Time series boosting models and sequence analysis may be appropriate for the sequence of representations, in particular those emphasizing the velocity and acceleration of the embeddings with respect to their relative directions to the targets.

A Additional Material and Experimental Results

A.1 Bake Off Supplements

Figure 45 supplements the smaller critical difference plot in Chapter 5. There exists a wide-spanning clique of the best models as measured by critical difference between average rank. This clique includes representation from each major approach to multivariate time series representation, discussed in Chapter 5. Chapter 6 proposes the new *ChemTime* model which belongs to the high-performing clique, but with substantially improved response time for inference and real-world real-time classification.

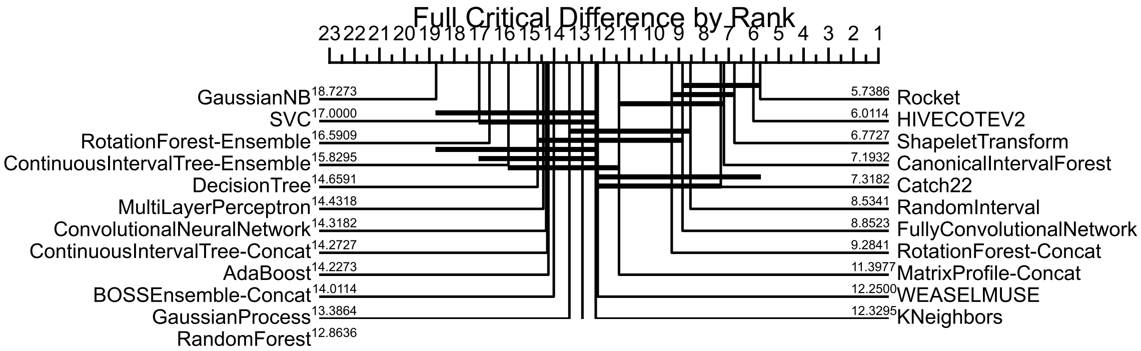


Figure 45: Critical difference plot demonstrating cliques of top 10 models by rank. Cliques demonstrate statistical certainty of difference in average rank after ranking by F1 score over 44 data set splits.

References

Bagnall, A., Dau, H. A., Lines, J., Flynn, M., Large, J., Bostrom, A., Southam, P., and Keogh, E. (2018). The uea multivariate time series classification archive, 2018.

Bagnall, A., Lines, J., Bostrom, A., Large, J., and Keogh, E. (2017). The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data mining and knowledge discovery*, 31(3):606–660.

Baxter, J. (2000). A model of inductive bias learning. *Journal of artificial intelligence research*, 12:149–198.

Bengio, Y., Courville, A., and Vincent, P. (2012). Representation learning: A review and new perspectives.

Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828.

Bostrom, A. and Bagnall, A. (2017). Binary shapelet transform for multiclass time series classification. In *Transactions on Large-Scale Data-and Knowledge-Centered Systems XXXII*, pages 24–46. Springer.

Bousmalis, K. and Levine, S. (2017). Closing the simulation-to-reality gap for deep robotic learning.

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (2017). *Classification and regression trees*. Routledge.

Brock, A., Donahue, J., and Simonyan, K. (2018). Large scale gan training for high fidelity natural image synthesis.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners.

Cai, C., Guo, P., Zhou, Y., Zhou, J., Wang, Q., Zhang, F., Fang, J., and Cheng, F. (2019). Deep learning-based prediction of drug-induced cardiotoxicity. *Journal of chemical information and modeling*, 59(3):1073–1084.

Caruana, R. (1997). Multitask learning. *Machine learning*, 28(1):41–75.

Chan, T. F., Golub, G. H., and LeVeque, R. J. (1982). Updating formulae and a pairwise algorithm for computing sample variances. In *COMPSTAT 1982 5th Symposium held at Toulouse 1982*, pages 30–41. Springer.

Chen, H., Engkvist, O., Wang, Y., Olivecrona, M., and Blaschke, T. (2018). The rise of deep learning in drug discovery. *Drug discovery today*, 23(6):1241–1250.

Child, R. (2021). Very deep vaes generalize autoregressive models and can outperform them on images.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297.

Cox, D. R. (1958). The regression analysis of binary sequences. *Journal of the Royal Statistical Society: Series B (Methodological)*, 20(2):215–232.

Dai, B. and Lin, D. (2017). Contrastive learning for image captioning. *arXiv preprint arXiv:1710.02534*.

David, L., Thakkar, A., Mercado, R., and Engkvist, O. (2020). Molecular representations in ai-driven drug discovery: a review and practical guide. *Journal of Cheminformatics*, 12(1):56.

Dempster, A., Petitjean, F., and Webb, G. I. (2020). Rocket: exceptionally fast and accurate time series classification using random convolutional kernels. *Data Mining and Knowledge Discovery*, 34(5):1454–1495.

Deng, H., Runger, G., Tuv, E., and Vladimir, M. (2013). A time series forest for classification and feature extraction. *Information Sciences*, 239:142–153.

DeVries, T., Romero, A., Pineda, L., Taylor, G. W., and Drozdzal, M. (2019). On the evaluation of conditional gans. *arXiv preprint arXiv:1907.08175*.

Donahue, J., Krähenbühl, P., and Darrell, T. (2016). Adversarial feature learning. *arXiv preprint arXiv:1605.09782*.

Elton, D. C., Boukouvalas, Z., Fuge, M. D., and Chung, P. W. (2019). Deep learning for molecular generation and optimization - a review of the state of the art. *CoRR*, abs/1903.04388.

Fix, E. and Hodges, J. L. (1989). Discriminatory analysis. nonparametric discrimination: Consistency properties. *International Statistical Review/Revue Internationale de Statistique*, 57(3):238–247.

Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139.

Frome, A., Corrado, G. S., Shlens, J., Bengio, S., Dean, J., Ranzato, M. A., and Mikolov, T. (2013). Devise: A deep visual-semantic embedding model. In Burges, C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Gebauer, N. W. A., Gastegger, M., Hessmann, S. S. P., Müller, K.-R., and Schütt, K. T. (2022). Inverse design of 3d molecular structures with conditional generative neural networks. *Nature Communications*, 13(1).

Goh, G. B., Siegel, C., Vishnu, A., Hodas, N. O., and Baker, N. (2017). Chemception: A deep neural network with minimal chemistry knowledge matches the performance of expert-developed qsar/qspr models.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial networks. *arXiv preprint arXiv:1406.2661*.

Gupta, A., Gupta, H. P., Biswas, B., and Dutta, T. (2020). Approaches and applications of early classification of time series: A review. *IEEE Transactions on Artificial Intelligence*, 1(1):47–61.

Hartvigsen, T., Sen, C., Kong, X., and Rundensteiner, E. (2019). Adaptive-halting policy network for early classification. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 101–110.

He, G., Zhao, W., and Xia, X. (2020). Confidence-based early classification of multivariate time series with multiple interpretable rules. *Pattern Analysis and Applications*, 23(2):567–580.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium. *arXiv preprint arXiv:1706.08500*.

Ho, T. K. (1995). Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE.

Hsu, E.-Y., Liu, C.-L., and Tseng, V. S. (2019). Multivariate time series early classification with interpretability using deep learning and attention mechanism. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 541–553. Springer.

Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., and Muller, P.-A. (2019). Deep learning for time series classification: a review. *Data mining and knowledge discovery*, 33(4):917–963.

Jaeger, S., Fulle, S., and Turk, S. (2017). Mol2vec: Unsupervised machine learning approach with chemical intuition. *ChemRxiv*.

Jönsson, P. and Eklundh, L. (2004). Timesat—a program for analyzing time-series of satellite sensor data. *Computers & geosciences*, 30(8):833–845.

Karras, T., Aittala, M., Hellsten, J., Laine, S., Lehtinen, J., and Aila, T. (2020). Training generative adversarial networks with limited data. *arXiv preprint arXiv:2006.06676*.

Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes.

Krizhevsky, A., Hinton, G., et al. (2009). Learning multiple layers of features from tiny images. *a*.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105.

Larochelle, H., Erhan, D., and Bengio, Y. (2008). Zero-data learning of new tasks. In *Zero-data Learning of New Tasks.*, volume 2, pages 646–651.

Li, L., Chang, Q., Xiao, G., and Ambani, S. (2011). Throughput bottleneck prediction of manufacturing systems using time series analysis. *Journal of Manufacturing Science and Engineering*, 133(2).

Lin, J., Keogh, E., Fu, A., and Van Herle, H. (2005). Approximations to magic: Finding unusual medical time series. In *18th IEEE Symposium on Computer-Based Medical Systems (CBMS'05)*, pages 329–334. IEEE.

Löning, M., Bagnall, A., Ganesh, S., Kazakov, V., Lines, J., and Király, F. J. (2019). sktime: A unified interface for machine learning with time series. *arXiv preprint arXiv:1909.07872*.

Lubba, C. H., Sethi, S. S., Knaute, P., Schultz, S. R., Fulcher, B. D., and Jones, N. S. (2019). catch22: Canonical time-series characteristics. *Data Mining and Knowledge Discovery*, 33(6):1821–1852.

Madan, R. and Mangipudi, P. S. (2018). Predicting computer network traffic: A time series forecasting approach using dwt, arima and rnn. In *2018 Eleventh International Conference on Contemporary Computing (IC3)*, pages 1–5.

Mater, A. C. and Coote, M. L. (2019). Deep learning in chemistry. *Journal of Chemical Information and Modeling*, 59(6):2545–2559.

Middlehurst, M., Large, J., and Bagnall, A. (2020). The canonical interval forest (cif) classifier for time series classification. In *2020 IEEE international conference on big data (big data)*, pages 188–195. IEEE.

Middlehurst, M., Large, J., Flynn, M., Lines, J., Bostrom, A., and Bagnall, A. (2021). Hive-cote 2.0: a new meta ensemble for time series classification. *Machine Learning*, 110(11):3211–3243.

Mirza, M. and Osindero, S. (2014). Conditional generative adversarial nets.

Moore, A. M., Paffenroth, R. C., Ngo, K. T., and Uzarski, J. R. (2023). Chemvise: Maximizing out-of-distribution chemical detection with the novel application of zero-shot learning. *arXiv preprint arXiv:2302.04917*.

Morgan, H. L. (1965). The generation of a unique machine description for chemical structures-a technique developed at chemical abstracts service. *Journal of chemical documentation*, 5(2):107–113.

Mori, U., Mendiburu, A., Keogh, E., and Lozano, J. A. (2017). Reliable early classification of time series based on discriminating the classes over time. *Data mining and knowledge discovery*, 31(1):233–263.

Mueller, T., Kusne, A. G., and Ramprasad, R. (2016). *Machine Learning in Materials Science*, chapter 4, pages 186–273. John Wiley and Sons, Ltd.

Nallon, E. C., Schnee, V. P., Bright, C. J., Polcha, M. P., and Li, Q. (2016). Discrimination enhancement with transient feature analysis of a graphene chemical sensor. *Analytical chemistry*, 88(2):1401–1406.

Nix, R. (2022). *Surface Science (Nix)*, chapter 1. Queen Mary, University of London.

Odena, A., Olah, C., and Shlens, J. (2017). Conditional image synthesis with auxiliary classifier gans.

Ottervanger, G., Baratchi, M., and Hoos, H. H. (2021). Multietsc: automated machine learning for early time series classification. *Data Mining and Knowledge Discovery*, 35(6):2602–2654.

Park, J., Groves, W. A., and Zellers, E. T. (1999). Vapor recognition with small arrays of polymer-coated microsensors. a comprehensive analysis. *Analytical chemistry*, 71(17):3877–3886.

Parrish, N., Anderson, H. S., Gupta, M. R., and Hsiao, D. Y. (2013). Classifying with confidence from incomplete information. *The Journal of Machine Learning Research*, 14(1):3561–3589.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Polikar, R. (2006). Ensemble based systems in decision making. *IEEE Circuits and systems magazine*, 6(3):21–45.

Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, 1(1):81–106.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. (2021). Learning transferable visual models from natural language supervision.

Radford, A., Metz, L., and Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. (2022). Hierarchical text-conditional image generation with clip latents.

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. (2021). Zero-shot text-to-image generation.

Rana, M., Chowdhury, P., et al. (2021). Modern applications of quantum dots: Environmentally hazardous metal ion sensing and medical imaging. In *Handbook of Nanomaterials for Sensing Applications*, pages 465–503. Elsevier.

Rao, C. R. (1948). Large sample tests of statistical hypotheses concerning several parameters with applications to problems of estimation. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 44, pages 50–57. Cambridge University Press.

Rifaioglu, A. S., Nalbat, E., Atalay, V., Martin, M. J., Cetin-Atalay, R., and Doğan, T. (2020). Deepscreen: high performance drug–target interaction prediction with convolutional neural networks using 2-d structural compound representations. *Chem. Sci.*, 11:2531–2557.

Rodriguez, J. J., Kuncheva, L. I., and Alonso, C. J. (2006). Rotation forest: A new classifier ensemble method. *IEEE transactions on pattern analysis and machine intelligence*, 28(10):1619–1630.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. (2021). High-resolution image synthesis with latent diffusion models.

Ruder, S. (2017). An overview of multi-task learning in deep neural networks.

Ruiz, A. P., Flynn, M., Large, J., Middlehurst, M., and Bagnall, A. (2021). The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery*, 35(2):401–449.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, 323(6088):533–536.

Rußwurm, M., Lefevre, S., Courty, N., Emonet, R., Körner, M., and Tavenard, R. (2019). End-to-end learning for early classification of time series. *arXiv preprint arXiv:1901.10681*.

Ryu, J. Y., Kim, H. U., and Lee, S. Y. (2018). Deep learning improves prediction of drug–drug and drug–food interactions. *Proceedings of the National Academy of Sciences*, 115(18):E4304–E4311.

Sauer, A., Schwarz, K., and Geiger, A. (2022). Stylegan-xl: Scaling stylegan to large diverse datasets.

Schäfer, P. (2015). The boss is concerned with time series classification in the presence of noise. *Data Mining and Knowledge Discovery*, 29(6):1505–1530.

Schäfer, P. and Leser, U. (2017). Multivariate time series classification with weasel+ muse. *arXiv preprint arXiv:1711.11343*.

Schmidt, J., Marques, M. R. G., Botti, S., and Marques, M. A. L. (2019). Recent advances and applications of machine learning in solid-state materials science. *npj Computational Materials*, 5(1):83.

Seeger, M. (2004). Gaussian processes for machine learning. *International journal of neural systems*, 14(02):69–106.

Semenick Alam, I. M. and Sickles, R. C. (2000). Time series analysis of deregulatory dynamics and technical efficiency: the case of the us airline industry. *International Economic Review*, 41(1):203–218.

Shinohara, Y. (2016). Adversarial multi-task learning of deep neural networks for robust speech recognition. In *INTERSPEECH*.

Silverman, B. W. and Jones, M. C. (1989). E. fix and jl hodges (1951): An important contribution to nonparametric discriminant analysis and density estimation: Commentary on fix and hodges (1951). *International Statistical Review/Revue Internationale de Statistique*, pages 233–238.

Socher, R., Ganjoo, M., Sridhar, H., Bastani, O., Manning, C. D., and Ng, A. Y. (2013). Zero-shot learning through cross-modal transfer.

Soloveitchik, M., Diskin, T., Morin, E., and Wiesel, A. (2021). Conditional frechet inception distance. *arXiv preprint arXiv:2103.11521*.

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society: Series B (Methodological)*, 36(2):111–133.

Suman Ravuri, O. V. (2019). Seeing is not necessarily believing: Limitations of biggans for data augmentation. In *International Conference on Learning Representations Workshop 2019*.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2015). Rethinking the inception architecture for computer vision.

Tenenbaum, J. B., de Silva, V., and Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323.

Tomchenko, A. A., Harmer, G. P., Marquis, B. T., and Allen, J. W. (2003). Semiconducting metal oxide sensor array for the selective detection of combustion gases. *Sensors and Actuators B: Chemical*, 93(1-3):126–134.

Wang, Z., Yan, W., and Oates, T. (2017). Time series classification from scratch with deep neural networks: A strong baseline. In *2017 International joint conference on neural networks (IJCNN)*, pages 1578–1585. IEEE.

Wei, J., Chu, X., Sun, X.-Y., Xu, K., Deng, H.-X., Chen, J., Wei, Z., and Lei, M. (2019). Machine learning in materials science. *InfoMat*, 1(3):338–358.

Weininger, D. (1988). Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci.*, 28(1):31–36.

Weiss, K., Khoshgoftaar, T. M., and Wang, D. (2016). A survey of transfer learning. *Journal of Big data*, 3(1):1–40.

Weiss, M., Wiederoder, M. S., Paffenroth, R. C., Nallon, E. C., Bright, C. J., Schnee, V. P., McGraw, S., Polcha, M., and Uzarski, J. R. (2018). Applications of the kalman filter to chemical sensors for downstream machine learning. *IEEE Sensors Journal*, 18(13):5455–5463.

Wiederoder, M. S., Nallon, E. C., Weiss, M., McGraw, S. K., Schnee, V. P., Bright, C. J., Polcha, M. P., Paffenroth, R., and Uzarski, J. R. (2017). Graphene nanoplatelet-polymer chemiresistive sensor arrays for the detection and discrimination of chemical warfare agent simulants. *ACS sensors*, 2(11):1669–1678.

Wiederoder, M. S., Weiss, M., Yoon, B., Paffenroth, R. C., McGraw, S. K., and Uzarski, J. R. (2019). Impact of graphene nanoplatelet concentration and film thickness on vapor detection for polymer based chemiresistive sensors. *Current Applied Physics*, 19(9):978–983.

Wiens, J., Horvitz, E., and Guttag, J. (2012). Patient risk stratification for hospital-associated c. diff as a time-series classification task. *Advances in Neural Information Processing Systems*, 25.

Wiswesser, W. J. (1968). 107 years of line-formula notations (1861-1968). *Journal of Chemical Documentation*, 8(3):146–150.

Xing, Z., Pei, J., and Yu, P. S. (2012). Early classification on time series. *Knowledge and information systems*, 31(1):105–127.

Yang, M., Tao, B., Chen, C., Jia, W., Sun, S., Zhang, T., and Wang, X. (2019). Machine learning models based on molecular fingerprints and an extreme gradient boosting method lead to the discovery of jak2 inhibitors. *Journal of Chemical Information and Modeling*, 59(12):5002–5012.

Yaqoob, U. and Younis, M. I. (2021). Chemical gas sensors: Recent developments, challenges, and the potential of machine learning—a review. *Sensors*, 21(8):2877.

Yeh, C.-C. M., Zhu, Y., Ulanova, L., Begum, N., Ding, Y., Dau, H. A., Zimmerman, Z., Silva, D. F., Mueen, A., and Keogh, E. (2018). Time series joins, motifs, discords and shapelets: a unifying view that exploits the matrix profile. *Data Mining and Knowledge Discovery*, 32(1):83–123.

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. (2017). mixup: Beyond empirical risk minimization.

Zhang, H., Zhang, Z., Odena, A., and Lee, H. (2019). Consistency regularization for generative adversarial networks. *arXiv preprint arXiv:1910.12027*.

Zhang, W., Wang, L., Chen, J., Xiao, W., and Bi, X. (2021). A novel gas recognition and concentration detection algorithm for artificial olfaction. *IEEE Transactions on Instrumentation and Measurement*.

Zhao, B., Lu, H., Chen, S., Liu, J., and Wu, D. (2017). Convolutional neural networks for time series classification. *Journal of Systems Engineering and Electronics*, 28(1):162–169.

Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232.